Artificial Intelligence in the Life Sciences 1 (2021) 100004

Contents lists available at ScienceDirect

Artificial Intelligence in the Life Sciences

journal homepage: www.elsevier.com/locate/ailsci

Quantifying sources of uncertainty in drug discovery predictions with probabilistic models

Stanley E. Lazic[a,*], Dominic P. Williams[b]

[a] Prioris.ai Inc., 459–207 Bank Street, Ottawa, K2P 2N2, Canada
[b] Functional and Mechanistic Safety, Clinical Pharmacology and Safety Sciences, AstraZeneca, R&D, Cambridge CB4 0WG, UK

ARTICLE INFO

Keywords:
Bayesian
Prediction
Probabilistic machine learning
Toxicity
Uncertainty quantification

ABSTRACT

Knowing the uncertainty in a prediction is critical when making expensive investment decisions and when patient safety is paramount, but machine learning (ML) models in drug discovery typically only provide a single best estimate and ignore all sources of uncertainty. Predictions from these models may therefore be over-confident, which can put patients at risk and waste resources when compounds that are destined to fail are further developed. Probabilistic predictive models (PPMs) can incorporate all sources of uncertainty and they return a distribution of predicted values that represents the uncertainty in the prediction. We describe seven sources of uncertainty in PPMs: data, distribution function, mean function, variance function, link function(s), parameters, and hyperparameters. We use toxicity prediction as a running example, but the same principles apply for all prediction models. The consequences of ignoring uncertainty and how PPMs account for uncertainty are also described. We aim to make the discussion accessible to a broad non-mathematical audience. Equations are provided to make ideas concrete for mathematical readers (but can be skipped without loss of understanding) and code is available for computational researchers (https://github.com/stanlazic/ML_uncertainty_quantification).

1. Introduction

At each stage of the drug discovery pipeline, researchers decide to progress or halt compounds using both qualitative judgements and quantitative methods. This is formally a prediction problem, where, given some information, a prediction is made about a future observable outcome. Standard predictive or machine learning models such as random forests, support vector machines, or neural networks only report point-estimates, or a single "best" value for a prediction; they provide no information on the uncertainty of the prediction. Prediction uncertainty is important when (1) the range of plausible values is as important as the best estimate, (2) users need to know that the model cannot confidently make a prediction, (3) users need to reliability distinguish between ranked items, or (4) the cost of an incorrect decision is large; for example, when making expensive investment decisions or when assessing patient safety.

To illustrate the importance of prediction uncertainty, Fig. 1 shows predicted clinical blood alanine aminotransferase (ALT) levels – an indicator of liver toxicity – for two hypothetical compounds. Assume that levels below 8 (arbitrary units) are considered safe, and that only one compound can be taken forward for clinical trials. Based only on the best estimate, compound A (blue) is preferable (Fig. 1A). Knowing the prediction uncertainty changes the picture (Fig. 1B). 14% of compound A's distribution is in the unsafe shaded region, while only 2% of compound B's distribution is in the unsafe region. Based on these distributions, compound B maybe the better candidate to progress to clinical trials; or a project team may decide to run more experiments to reduce compound A's uncertainty.

We define a probabilistic predictive model (PPM) as any machine learning model that returns a distribution for the prediction instead of a single value. PPMs differ in how they represent uncertainty. At one end, fully probabilistic Bayesian models specify a distribution for the outcome and for all unknown parameters in the model. These models are the gold-standard for quantifying uncertainty but they can be computationally expensive. At the other end are methods that return multiple predicted values without specifying a probability distribution. Examples include fitting the same model on multiple bootstrapped datasets, or for models with a stochastic component, fitting the same model using different random number generator seeds [31]. Although these models are often computationally tractable, the connection between the distribution of predicted values and uncertainty is unclear. For example, does varying something trivial such as the random number generator seed adequately capture our uncertainty in a prediction? Between these extremes are approaches that try to obtain the benefits of fully probabilistic models using approximations, reformulations, or computational shortcuts. For example, instead of using a fully Bayesian deep neural

[*] Corresponding author.
E-mail address: stan.lazic@cantab.net (S.E. Lazic).

https://doi.org/10.1016/j.ailsci.2021.100004
Received 17 May 2021; Received in revised form 24 May 2021; Accepted 25 May 2021
Available online 26 May 2021

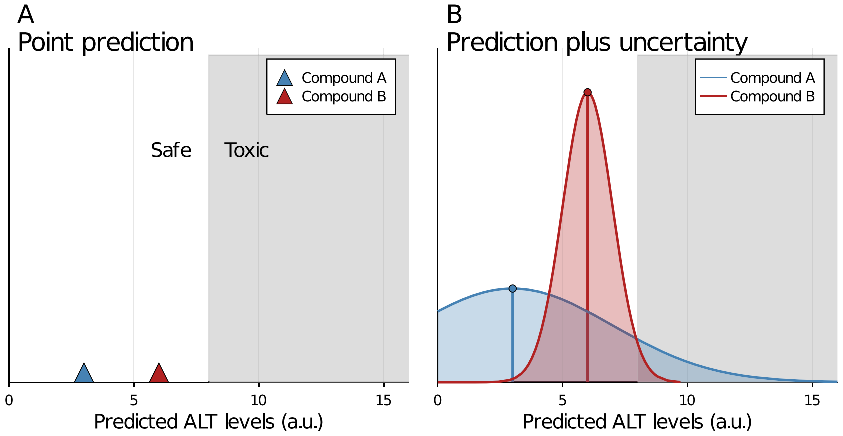

Fig. 1. Prediction uncertainty. Predicted blood ALT levels for two compounds, with Compound A appearing better (A). Compounds with values above the safety threshold of 8 (grey shaded region) will not be progressed. Reporting prediction uncertainty shows that 18% of Compound A is above the threshold but only 2% of Compound B, indicating that Compound B is better (B). ALT = alanine aminotransferase; a.u. = arbitrary units.

network, Kristiadi et al. were able to obtain many of the benefits by making only the final layer of the network Bayesian [29]. Also in the neural network literature, Gal and Ghahramani approximated parameter uncertainty by randomly inactivating nodes at prediction time – a procedure known as Monte Carlo dropout [15], and Teye and colleagues used the mean and variances calculated from batch normalisation steps for the same purpose [63]. These approximate methods are an active area of research [3,27,46,70] (see Mervin et al., for a recent review [40]), and will be critical for making PPMs more widely adopted, but here we focus on fully probabilistic models as they better highlight the key areas of uncertainty.

All prediction models can be written as

$$y \mid x \tag{1}$$

and read as "y given x" – where y is the outcome to be predicted and x are one or more variables used to predict y. Both x and y are available when training a model, and when a model is deployed, new x values are observed and used to predict the unknown y's. PPMs additionally provide a probability distribution for y, denoted as

$$P(y \mid x) \tag{2}$$

and read as "the probability of y given x". A distribution for y enables us to calculate any metric of interest, such as the best guess for y (e.g. mean, median, or mode), thereby providing the same information as standard methods. But in addition, prediction intervals (PI) can be calculated around the best estimate, or the probability that y is greater or less than a predefined threshold can be calculated, as was done in Fig. 1B.

Below we describe the sources of uncertainty and the advantages of PPMs. We aim to make the discussion accessible to a broad non-mathematical audience. Equations are provided to make ideas concrete for mathematical readers, but can be skipped without loss of understanding of the remaining text. In addition, code is provided for computational researchers (https://github.com/stanlazic/ML_uncertainty_quantification) and implemented in Julia using Turing [1,16].

2. Sources of uncertainty

The definition of a probabilistic model in Eq. (2) lacked a crucial component, which is the model itself:

$$P(y \mid x, \text{Model}). \tag{3}$$

The sources of uncertainty are now clear: they can reside in the data (x, y) or in the model. A prediction for y not only depends on x, but on the model used to connect x and y. Hence, predictions are conditional on a model, and uncertainty in the model should lead to greater uncertainty in the prediction. Models are composed of the following components, which we discuss in greater detail below:

1. **Data.** The outcome y and the predictor or input variables x.
2. **Distribution function.** The distribution that represents our uncertainty in y, also called the likelihood or data generating distribution: $G(\cdot)$.
3. **Mean function.** The functional or structural form of the model describing how y changes as x changes: $f_\mu(\cdot)$.
4. **Variance function.** Describes how the uncertainty in y varies with x: $f_\sigma(\cdot)$.
5. **Parameters.** The unknown coefficients or weights for the mean (θ_μ) and variance (θ_σ) functions that are estimated from the data.
6. **Hyperparameters.** Parameters or other options used to define a model that are not estimated from the data but fixed or selected by the analyst: ϕ.
7. **Link functions.** Nonlinear transformations of the mean ($l_\mu(\cdot)$) and/or variance function ($l_\sigma(\cdot)$) used to keep values within an allowable range.

Combining these seven components gives the generic formulation of a PPM (Eq. (4)). Although this equation is abstract, it captures where uncertainty can reside. In this section we carefully describe the terms in the equation and then provide a concrete example.

$$\begin{aligned} y &\sim G(\mu, \sigma) \\ \mu &= l_\mu(f_\mu(x; \theta_\mu)) \\ \sigma &= l_\sigma(f_\sigma(x; \theta_\sigma)) \end{aligned} \tag{4}$$

Starting with the first line of Eq. (4), y is a future value that we want to predict and it could represent a clinical outcome, an IC_{50} value, or a physicochemical property of a compound such as solubility. PPMs require that we specify a distribution for our uncertainty in y, which we can informally think of as the distribution from which y was generated. We denote this distribution as $G(\cdot)$ and the "$(\cdot)$" notation indicates that G is a function with inputs (μ and σ in this case), but places the focus on G and not the inputs, thereby reducing clutter. Common distributions include the normal/Gaussian (for continuous symmetric data), Student-t (continuous symmetric data with outliers), Bernoulli (0/1 data), and Poisson (count data). The mean of the chosen distribution is given by μ and many distributions have a second parameter, σ, that controls the spread or width of the distribution. This parameter is critical for PPMs because it describes the uncertainty in y. The $\sim$ symbol can be read as "is distributed as" or "is generated from". To make this concrete, we might represent our uncertainty in y for a given compound with a Gaussian distribution that has a mean $\mu = 2.45$ – which represents our best estimate of y – and a standard deviation of $\sigma = 2.1$. This would be written as $y \sim \text{Normal}(2.45, 2.1)$.

But where did our best estimate μ come from? This is defined on the second line of Eq. (4). x is the input data used to predict y and it could represent compound structures (encoded as a binary fingerprint for example), assay results, or physicochemical properties. The prediction task reduces to using x to predict y, but they need to be connected through a statistical or machine learning model. The structural or functional form of this model is denoted by $f_\mu(\cdot)$ and it could represent the structure of a simple linear regression model, the architecture of a neural network, an ensemble of trees, or a differential equation representing a pharmacokinetic model. We refer to $f_\mu(\cdot)$ as the *mean function* because it tells us the predicted mean value of y for a given value of x. The mean function contains parameters (θ_μ) that are estimated from the data, and the parameters could represent the coefficients of a linear model or the weights and biases of a neural network. The "learning" in machine learning refers to finding parameter values that maximise predictive performance, given the data and functional form of the model. θ_μ usually represents multiple parameters in the mean function; for example, it would represent both the intercept and slope in a simple regression model. The subscript μ on f and θ indicates that the function and parameters refer to the mean, since we also have a function and parameters for the variance, σ, which we described further below.

A potential problem is that μ is unconstrained and can be any value calculated from $f_\mu(\cdot)$, which may lead to impossible predictions. For example, if we're predicting the probability that a compound is toxic, $f_\mu(\cdot)$ needs to be between zero and one – values outside this range do not make sense. Hence, we need to transform $f_\mu(\cdot)$ with a *link function* $l_\mu(\cdot)$ to put it within a permissible range. One option is to use the equation $1/(1+\exp(-f_\mu(\cdot)))$ as a link function, which compresses $f_\mu(\cdot)$ into the 0–1 range, but other functions are also possible. A link function is unnecessary when no restriction on $f_\mu(\cdot)$ is required, and $l_\mu(\cdot)$ can be dropped from the formula. (Useful mnemonics: f = Function; G = data Generating distribution; l = Link function; with subscripts μ and σ referring to the mean and variance, respectively).

The third line of Eq. (4) shows that the variance or uncertainty in a predicted value of y can be specified in the same way as we specify the mean of y. Many models assume a constant value for σ, which is the "homogeneity of variance" assumption in traditional statistical models. However, a model's uncertainty in y may vary for different values of x, which can be captured with a *variance function* $f_\sigma(\cdot)$. Since variances are positive, a link function for the variance ($l_\sigma(\cdot)$) is needed to constrain σ to be greater than zero.

Fully Bayesian PPMs also require us to specify the uncertainty in the parameters θ_μ and θ_σ before analysing the data, and the hyperparameters (ϕ) refer to this specification. Many of the approximate methods avoid this step and ϕ may not correspond to any part of the model, but defines other options for the learning algorithm and therefore it is not included in Eq. (4).

Usually the x and y data are all we have, and we need to choose G, l_μ, f_μ, l_σ, f_σ, and ϕ based on background knowledge, preliminary plots of the data, or trying several options and empirically assessing which is best. For example, an initial model might assume that G is Gaussian and that f_μ follows a hypothesised mechanistic relationship based on a pharmacokinetic model. Then, we estimate or learn the values of θ_μ based on training data, and assess the prediction on a separate test data set. To make these ideas concrete, Fig. 2 shows simulated data for 100 compounds, where y is a clinical outcome and x is an assay result. Assume that higher values of y indicate greater toxicity.

A nice feature of PPMs is that they are generative, meaning that they can generate or simulate data. Indeed, simulation and learning are opposite sides of the same coin: learning takes the fixed data and infers likely values of the parameters that could have generated the data, whereas simulation fixes the parameters and generates the data. The model in Eq. (5) generated the data in Fig. 2 and we will use it as a running example throughout

$$\begin{aligned} y &\sim \text{Normal}(\mu, \sigma) \\ \mu &= \theta_2 + \frac{1 - e^{-\theta_1 x}}{1 + e^{-\theta_1 x}}. \end{aligned} \tag{5}$$

The first line of the equation is read as: "the outcome y is generated ($\sim$) from a Gaussian or Normal distribution with a mean of μ and a standard deviation of σ". But how does y depend on x? The second line shows how x enters and how it depends on two parameters: θ_1 and θ_2 (e is a constant, not a parameter). We set this equation equal to μ and can substitute it for μ in the first line of Eq. (5) giving

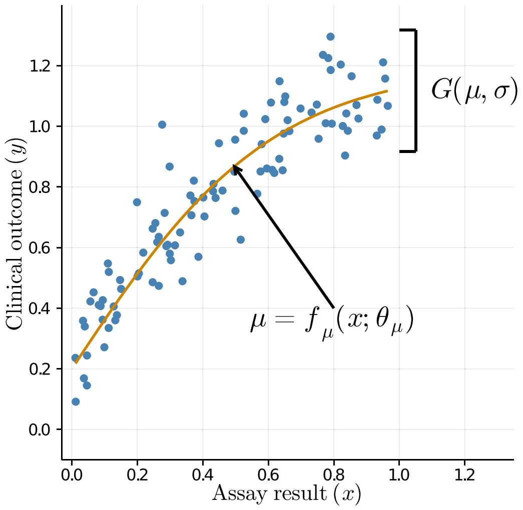

Fig. 2. Simulated data with the true relationship between x and y given by $\mu = f_\mu(x;\theta_\mu)$ (orange curve) and based on Eq. (5). G is the Gaussian data generating distribution with a mean μ and a constant variance σ, which models the spread of points around the line. Link functions for μ and σ are not used and hence are not shown.

$$y \sim \text{Normal}\left(\theta_2 + \frac{1 - e^{-\theta_1 x}}{1 + e^{-\theta_1 x}}\,\sigma\right).$$

Writing the equation in one line makes the relationship between x and y clearer, but multi-lined equations are easier to read with more complex models. For most prediction models the parameters are uninteresting, but to help interpret the model, θ_2 is the y-intercept (value of y when $x = 0$), and θ_1 controls how quickly the line in Fig. 2 reaches the upper asymptote as x gets large.

To simulate a value for y, we need to (1) select parameter values, and we use the following: $\theta_1 = 3.25$, $\theta_2 = 0.2$, and $\sigma = 0.1$; (2) select a value of x, which enables us to calculate μ; then (3) draw a random number from a Gaussian distribution with a mean of μ and standard deviation of σ. This can be repeated any number of times to obtain the $P(y|x)$ distribution, and for different values of x. The data in Fig. 2 were generated for 100 x values uniformly distributed between 0 and 1. Note that θ_μ in Eq. (4) is a place-holder for several variables, which correspond to θ_1 and θ_2 in Eq. (5). Given this data, we now illustrate were where the seven sources of uncertainty enter.

2.1. Mean function uncertainty

Uncertainty in the mean function $f_\mu(\cdot)$ arises because we rarely know the true form of the relationship between x (assay) and y (outcome) – that is, we don't know the form of Eq. (5). Uncertainty in the mean function is also called model uncertainty, but this term is ambiguous because models have multiple components. Choices for the mean function include which predictors, interaction terms, transformations, basis expansions, hierarchies, and time-varying components to include in the model. Assume we only observe the data in Fig. 2, several models we might consider for the relationship between x and y are:

$$\begin{aligned} \text{Linear}: &\quad y = \theta_0 + \theta_1 x \\ \text{Quadratic}: &\quad y = \theta_0 + \theta_1 x + \theta_2 x^2 \\ \text{2 Parameter Exponential}: &\quad y = \theta_2\ (1 - e^{-\theta_1 x}) \\ \text{3 Parameter Exponential}: &\quad y = \theta_3 + \theta_2\ (1 - e^{-\theta_1 x}) \\ \text{Michaelis} - \text{Menten}: &\quad y = \theta_1 x\ /\ (\theta_2 + x). \end{aligned}$$

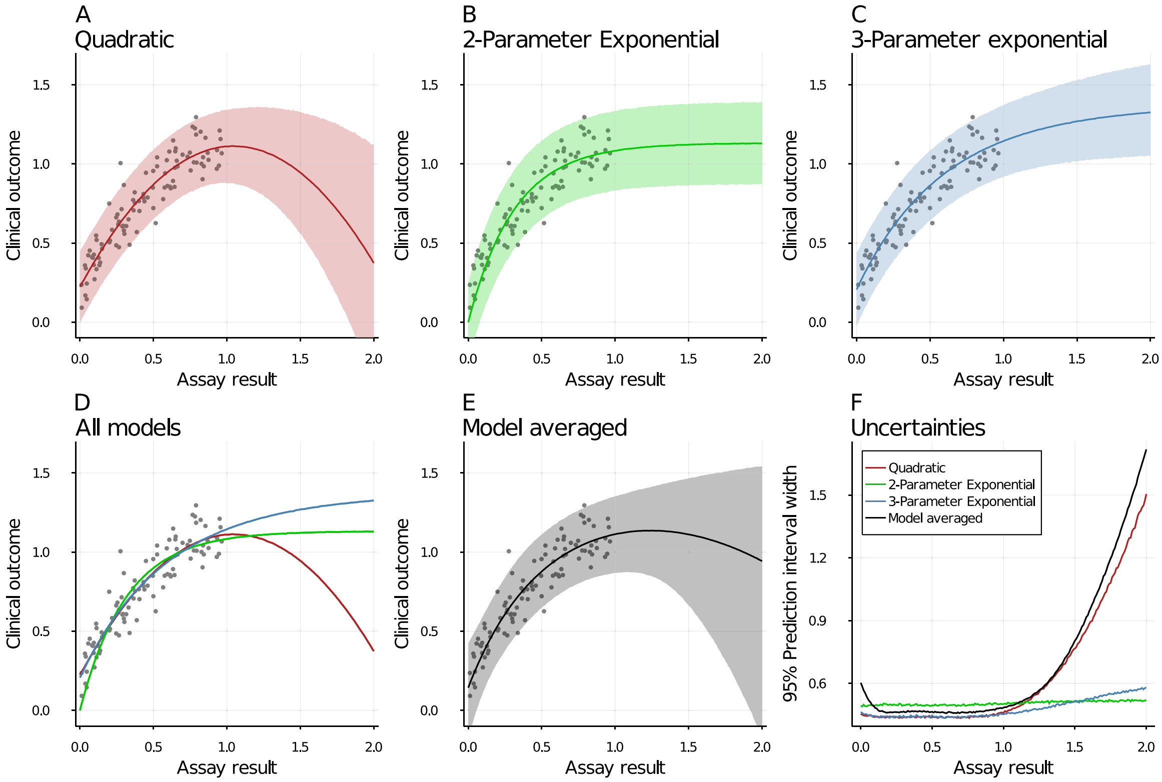

Fig. 3. Model averaging. Three models fit the data well (A-C), even though none are the true model. They make different predictions at low assay values and when extrapolating to higher values. The mean predictions are superimposed for easier comparison (D). Model averaged prediction (E). Comparison of prediction interval widths (F). Shaded regions are the 95% prediction intervals.

None of these are the true model (Eq. (5)), but except for the linear model, they all saturate at high values of x or are concave and therefore capture the main trend in the data. Which model should we use? Typically, only a single model is selected and predictions are made from that. If one model is clearly better than the others, there may be little lost by using one model for predictions. However, if two or more models fit the data equally well, making predictions from only one will underestimate the prediction uncertainty. Fortunately, we are not forced to choose one model but can fit several and combine their predictions. To illustrate, we will use the quadratic, 2-parameter exponential, and 3-parameter exponential models. The three models are fit to the data (Fig. 3A–C) and predictions are extrapolated to show both how similar the fits are where there is data, and how different the fits are when extrapolating. The shaded regions show the 95% prediction intervals (PI), and if a model is suitable, we expect 95% of the data to fall in the shaded region.

To better compare the predictions, the mean functions μ for three models are plotted together in Fig. 3D. The models make similar predictions within the range of the data, except at very low values of x, where the 2-parameter exponential model predicts smaller values. Fig. 3E shows the model-averaged prediction, and note how uncertainty is greater at high values of x, where the models make different predictions. To better appreciate how model averaging incorporates uncertainty, Fig. 3F shows the width of the 95% PI for the three models and the averaged model. Note how the averaged model has a wider PI at low values of x than any of the original models. This occurs because the three models make different predictions at low values and hence this region is more uncertain. For intermediate values of x, all three models make similar predictions and the averaged PI is wider than some models and lower than others. When extrapolating to larger values of x, the model averaged PI width quickly becomes the widest, reflecting both the diverging predictions of the individual models and the greater uncertainty in the quadratic model.

Predictions are always conditional on a model, and if we don't know the true model, predictions from the wrong model are likely to be overconfident. We extrapolated well beyond the data to illustrate how model averaging accounts for model uncertainty. Such extrapolation may seem unrealistic, but the message is that whenever models make different predictions for the same values of x, the uncertainty in the model-specific predictions will be overconfident, as we see to a lesser degree for low values of the assay.

2.2. Parameter uncertainty

Most predictive models have parameters or weights that are learned from the data and control how the predictor variables (x) are related to the outcome variable (y) – these parameters are the θ symbols in the five models considered above. Most machine learning methods only use the single best value of each parameter when making a prediction. But since the parameters are learned from the data, they are uncertain, and this uncertainty should be propagated into the prediction. Parameter uncertainty decreases as the sample sizes increases, so to better illustrate the effect of parameter uncertainty on predictions, a smaller dataset was made by taking every eighth data point from the previous example.

Fig. 4 uses the quadratic model, which has four parameters ($\theta_0, \theta_1, \theta_2, \sigma$). The grey shaded region shows the 95% prediction interval from a Bayesian model that accounts for parameter uncertainty. The dashed black lines show the 95% PI from a classic quadratic regression model which ignores parameter uncertainty, and note how they are slightly narrower. The mean function is identical for both the Bayesian and classic model.

The difference in PI width may seem negligible when focusing on the mean prediction, but ignoring parameter uncertainty gives approximately 7% narrower PIs. This may be important with "point of departure" calculations when the tails of the distributions are more important than the means [50]. Assume clinical outcomes above 1.2 are considered

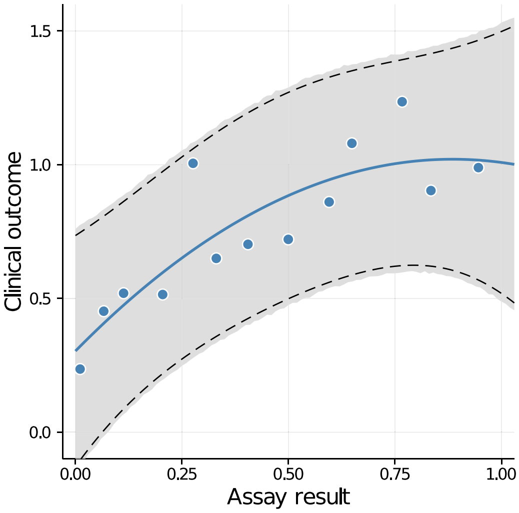

Fig. 4. Parameter uncertainty. The grey shaded region is the 95% prediction interval for a Bayesian model and the narrower black dashed lines for a classical model. The classic interval is narrower because it doesn't account for parameter uncertainty.

problematic and the assay result for one compound is $x = 0.5$. When fully accounting for parameter uncertainty, there is a 5.9% chance that the true value of the clinical outcome is above 1.2, versus a 4.3% chance when ignoring uncertainty. The ratio of these numbers is 1.37, indicating that the tail area is nearly 1.4 times larger when accounting for uncertainty.

Models typically have many more parameters than this example (the state-of-the-art Generative Pre-trained Transformer 3 (GPT-3) deep learning language model has 175 billion parameters [6]) and simply collecting more data is often not an option to reduce parameter uncertainty because more data enables more complex models to be fit (e.g. including nonlinear terms and interactions), which then increases the number of parameters.

2.3. Hyperparameter uncertainty

Hyperparameters are a diverse set of tunable options that affect the training and predictions. Unlike parameters, they are not estimated from the data but selected by the analyst; examples include the amount of regularisation in a lasso model, the number of trees in a random forest model, or the cost function in a support vector machine. Suitable values are typically found by trying several options and selecting the best using crossvalidation. In the hyperparameter category we can also include options that are rarely part of a formal selection process such as the choice of optimisation algorithm or random number seed for models with a stochastic component. For fully Bayesian models we can also include parameters for prior distributions, which are not updated by the data. These hyperparameters are selected pragmatically to provide good predictions, but other sets of hyperparameter values might give equally good predictions, on average, but slightly different predictions for each test compound. Hence, uncertainty in hyperparameter values is rarely taken into account. A further complication is that most hyperparameters are not related to any biological or chemical quantity of interest and hence it is unclear what the uncertainty is actually about. Nevertheless, Lakshminarayanan and colleagues showed that by running many models with a different random seed, the ensemble of predictions performed better than a single model, and the distribution of predicted values provided a measure of uncertainty [31].

2.4. Data uncertainty

One of the main sources of uncertainty – and which is almost universally ignored – is the uncertainty in the data, both in the predictors (x) and in the outcome to be predicted (y). The uncertainty can be in the training data used to build the model, in the new data to be predicted, or both. The main sources of data uncertainty are measurement error, misclassification error, binning, censoring and truncation, and missing values. Each of these are discussed below.

Predictors are often experimental measurements and are therefore subject to measurement error, or they are samples from a larger population and are therefore subject to sampling error (e.g. only cells in the field of view are measured, not all cells in a well, and if a different subset of cells were selected, a differnt measured value would be obtained). Predictors may also be calculated quantities such as IC_{50} values estimated from dose-response or concentration-response curves, and hence are uncertain. Furthermore, some predictors such as cLogP are the output of other (imperfect) prediction models and therefore are also uncertain. Finally, some predictors are not measured directly but are estimated from a standard curve, which introduces additional uncertainty because the curves may not be not perfectly calibrated.

All these are examples of classic measurement error, defined as $x_{\text{measured}} = x_{\text{true}} + \text{error}$, where the measured x value is the true value corrupted by some error or noise. Errors can also be multiplicative, where $x_{\text{measured}} = x_{\text{true}} \times \text{error}$. Another type of error is Berkson error, where samples or experimental units are assumed to have the same exposure but actually differ. For example, several wells in a microtitre plate are given the same concentration of a compound, but the true concentration may differ due to variations in the amount of compound dispensed, or, wells on the edge of a plate may have greater evaporation of the solution and thus have a higher effective concentration of the compound. Compound toxicity classifications can also introduce Berkson error. A compound may be classified as "severely" hepatotoxic, even though most people tolerate the compound well and only a few experience severe reactions. The class label is therefore defined by a few members of the class instead of the majority response.

Berkson error can be introduced when converting continuous values into bins or groups. For example, compounds are categorised as active versus inactive, despite having a range of activity values. Or, compounds are classified as having no, mild, or severe toxicity, even though compounds will have a range of toxicity levels *within* each category. Binning can also lead to misclassification error, where a compound is placed into the incorrect category. This can occur if the measured assay value differed from the true value and fell on the wrong side of a threshold. Hence, binning is strongly discouraged [33,35]. Misclassification can also occur due to incorrect diagnoses, labelling errors, or data-entry errors. The standard response to these known and often large sources of data uncertainty is to ignore them and assume that $x_{\text{measured}} = x_{\text{true}}$.

Ignoring error in x can bias parameter estimates, but for prediction models the parameters are usually not of interest. Even though noisy data can lead to biased parameter estimates, the model is still consistent for the prediction, meaning that as the sample size increases, the prediction will be correct, on average [9,20]. This likely explains why error in x has received little attention in the predictive modelling and machine learning literature. However, if we're interested in the uncertainty in the prediction, then making good predictions on average is not good enough, we need to ensure that the prediction uncertainty is calibrated.

Data are *censored* when they are known only up to a boundary value, but not beyond, and therefore only partial information is available. For example, assays typically have upper and lower limits of detection (LoD) and uncertainty arises because the exact value is unknown, but the LoD is typically treated as the "true" measured value.

Data are *truncated* when values outside of a range are omitted, and the number of omitted values is unknown. For example, for objects to be segmented as a cell in a standard image analysis, they must have a minimum user-defined cell size. Smaller cells will therefore not be included in the analysis, and hence both the estimated cell size and properties that are correlated with cell size can differ from their true values, thereby introducing both uncertainty and bias.

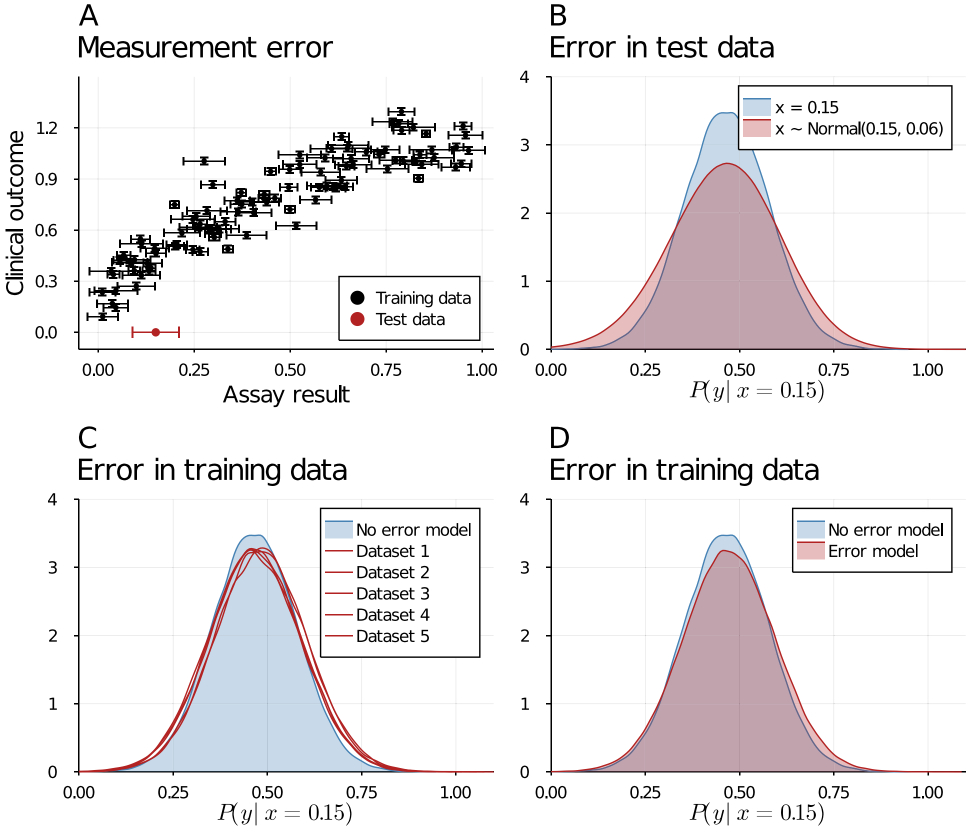

Fig. 5. Data uncertainty. Error bars indicate 1 standard error of the estimated assay value and clinical outcome (A). The red point at $x = 0.15$ is the new value to be predicted. Predictions are more uncertain when measurement error in the test data is included (assuming the training data is measured without error; B). Predictions for the test data when accounting for measurement error in the training data using multiple generated data (assuming no error in the test data; C). Averaged predictions from the generated data shows greater prediction uncertainty compared with ignoring measurement error in the training data (D). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Missing data is the final source of data uncertainty and can arise for many reasons. Imputation is a common approach to deal with missing data, where a plausible value is generated and substituted for the missing value. The imputed value is then taken as the true value, ignoring that it was generated and not measured. A simple way to account for uncertainty in an imputed value is to impute many values – known as multiple imputation – and the variation in the imputed values captures the uncertainty [37,65]. Predictions are the made for each imputed value and the predictions combined.

To illustrate data uncertainty, Fig. 5A shows the same simulated data but with uncertainty in both the predictor and outcome variable. Here we assume that the error in x differs for each sample because it depends on the precision of the measurement, whereas the uncertainty in y is constant; for example, we know that the measurements are accurate to ± some fixed amount. Variable and fixed uncertainty are accounted for in the same way, and we include both for illustration purposes.

The uncertainty in x and y can be handled in two ways. The first is to directly model the errors in a Bayesian analysis [20,39,41,51]. For example, if x is measured with error, the true value can be inferred with $x_{\text{measured}} \sim \text{Normal}(x_{\text{true}}, \sigma_{\text{error}})$ where σ_{error} is the uncertainty in the measured value of x – usually the standard error. However, fitting such models may be difficult as each sample (row) adds as many parameters as there are variables with measurement error (columns).

A second simpler option is to generate multiple data sets, where the x and y values are drawn from a distribution [2]. For example, suppose an x variable for one compound has a measured value of 0.5 with a standard error of ±0.06. We can sample, say, 10 values from a normal distribution with a mean of 0.5 and a standard deviation of 0.06, thereby generating 10 new datasets where the observed value of x is replaced with the generated value. This is a form of multiple imputation and Blackwell, Honaker, and King describe a more sophisticated method of generating new data by taking the correlations between variables into account [2]. Each dataset is then analysed separately, and the predictions from each analysis are combined. Variations between the different datasets will lead to different parameter estimates, which in turn will lead to different predictions, and the ensemble of predictions captures the uncertainty in x. The models can be fit to the separate datasets in parallel, and so the computation time is the same as fitting the model to one dataset.

We use the second method to illustrate the effect of ignoring measurement error in two ways. First, we assume the training data is measured without error, but the test data is measured with error. Then we assume that the training data is measured with error but the test data is not. In both cases we compare the result to the standard approach of ignoring measurement error in both the training and test data. The test data is a single new compound with an assay value of $x = 0.15 \pm 0.06$, shown as the red point in Fig. 5A (plotted at $y = 0$, but the objective is to predict y). The 3-parameter exponential model is used in this example. The narrow blue distribution in Fig. 5B corresponds to the standard approach of ignoring uncertainty in both the training and test data, and the red distribution shows the greater uncertainty in the prediction for y when the measurement error in x is included. To obtain this distribution, 1000 samples were drawn from a normal distribution with a mean of 0.15 and standard deviation of 0.06. A prediction was made for each of these 1000 samples which reflects the uncertainty in x.

The blue distribution in Fig. 5C is again the standard analysis, and the five red lines show the slightly different predictions from each of the five datasets that account for measurement error in the training data (the test data was assumed to be error-free). Predictions from the five datasets are averaged and shown as the red distribution in Fig. 5D, which is slightly wider than the standard analysis from the blue distribution. The additional uncertainty appears negligible in this example, especially compared with uncertainty in the test data (Fig. 5B). However, the variation between datasets is expected to increase with (1) greater uncertainty in the variables, (2) more variables with measurement error included in the model, and (3) more parameters in the model with the total sample size remaining fixed. The effect of measurement error in the training data can be assessed during model development and validation, and if the additional prediction uncertainty is negligible, then the final production model might ignore it.

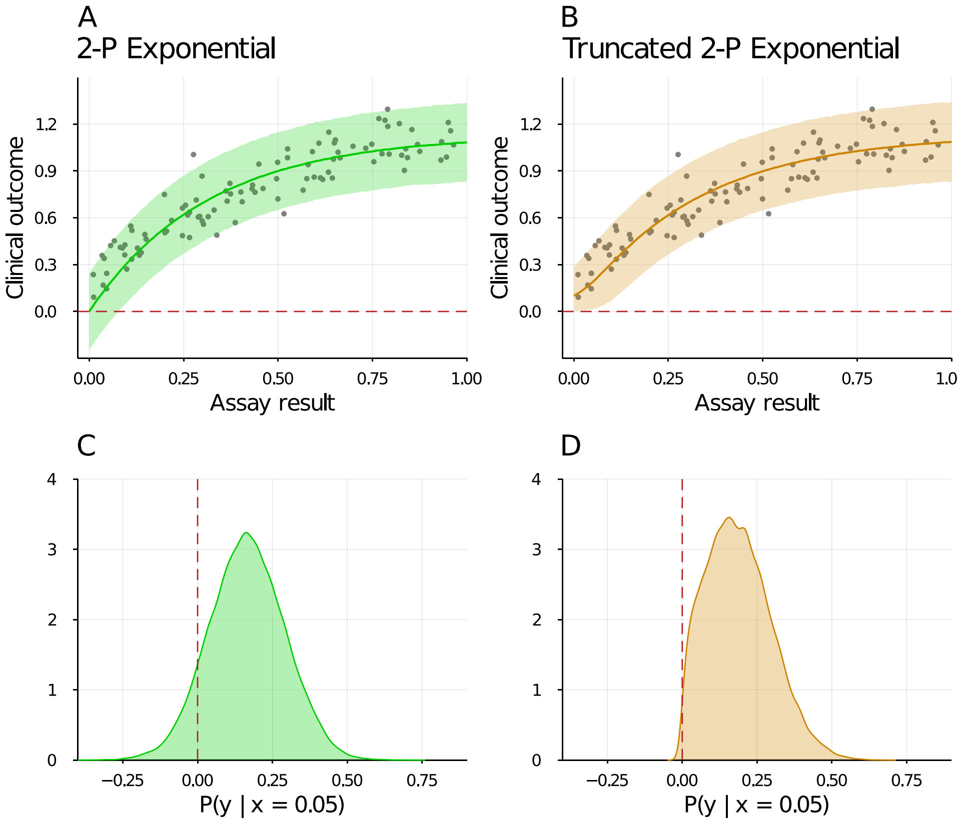

Fig. 6. Truncated predictive distributions. Fits and 95% PI for models without (A and C) and with (B and D) a constraint that the predicted values must be positive. Prediction for a new compound given $x = 0.5$ without the constraint shows that 9.1% of the predicted distribution is negative (C), while the truncated model redistributes the prediction to positive values (D). Red dashed lines are the data boundary.

2.5. Distribution function uncertainty

Another source of uncertainty is the distribution function $G(\cdot)$, which represents our uncertainty in a predicted value of y. Dozens of distributions are available but the list can be narrowed down based on background knowledge of the outcome. For example, if the outcome is binary such as absent/present, safe/toxic, or alive/dead, then a Bernoulli distribution is appropriate; if the outcome is a count such as the number of seizures, then a Poisson or negative binomial distribution are two common options; if the data are positive values and skewed such as liver enzyme levels, then a log-normal or gamma distribution may be suitable; if the outcome is an ordered category such as none/mild/severe, then an ordered categorical distribution would be appropriate; if the outcome is continuous and unbounded with no outliers, then a Gaussian distribution may be suitable; and if there are outliers, a Student-t distribution might be appropriate. Many Bayesian textbooks have appendices that list the common distributions and their properties [18,36,38].

Choosing between distributions is made easier because many distributions are special cases of other distributions. For example, both the Gaussian and Cauchy distributions are special cases of the Student-t distribution, the Poisson distribution is a special case of the negative binomial distribution, and the exponential distribution is a special case of a gamma distribution. Hence, we often don't need to choose between a set mutually exclusive options, but can select the more general distribution and allow the model to determine if one of the special cases is more appropriate. The more general distributions usually have only one additional parameter and therefore do not make the model much more complex. However, not all potentially suitable distributions are related (e.g. gamma and lognormal) and hence two or more models may need to be compared. The data for our running example was generated from a Gaussian distribution and which we have been using for all the models throughout. Hence using the more general Student-t distribution will inform us that the Gaussian is suitable, and so the results are not shown.

A key consideration when selecting a distribution function is the bounds of the data. In our running example, the clinical outcome has a minimum value of zero, but is being modelled with a Gaussian distribution. Since a Gaussian distribution is defined for both positive and negative numbers, there is nothing to prevent negative predictions. A model is clearly inappropriate if it predicts impossible values. Fortunately, we can easily define truncated versions of standard distributions, and so we could specify a Gaussian distribution with a lower bound of zero. Fig. 6 shows an example using the 2-parameter exponential model to predict the clinical outcome for an assay value of $x = 0.05$ both without (A) and with (B) truncation at $y = 0$. Without truncation the model gives 9% chance that y will be less than zero (Fig. 6C). With truncation, this probability gets redistributed to positive values (Fig. 6D, the small proportion of the distribution below zero is a plotting artefact). Hence, if training or test data are near boundaries, using truncated versions of standard distributions is sensible. However, if the data are bounded but the values are far from the boundaries, then accounting for such boundaries may be unnecessary.

2.6. Link function uncertainty

Link functions are required when the predicted mean $\mu = f_\mu(\cdot)$ is bounded by an upper and/or lower limit. For example, when predicting the probability of an event, the predicted value must lie between 0 and 1. Values returned from the mean function are unconstrained and can lie well outside this range. Hence, a link function is used to transform the values to respect the bounds. The logit, probit, cauchit, and complementary log-log functions all take unconstrained numbers and compress them into the 0–1 range (Fig. 7). There is no "correct" link function and each provides a different mapping from the unconstrained input to the constrained output and hence gives a different prediction, especially for large values of the input. Link functions are also required for the variances, since variances cannot be negative values, and exponential or power links are often used.

Link functions are analogous to activation functions in neural networks, although they are used as nonlinear transformations between neurons and not necessarily to constrain values to allowable ranges. But

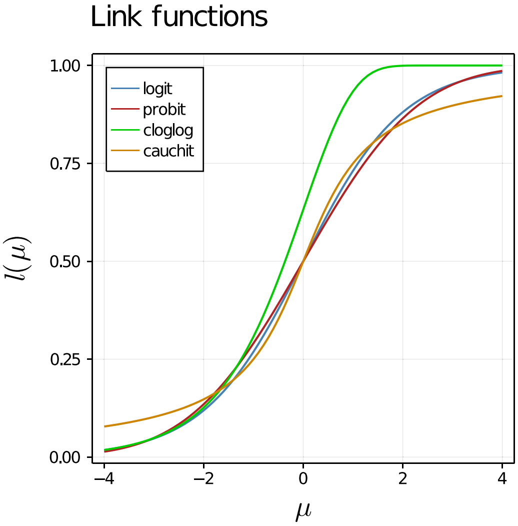

Fig. 7. Link function uncertainty. Four functions that map the mean function $\mu = f(\cdot)$ from $-\infty$ to ∞ to values between 0 and 1. These link functions are required when the outcome is a probability and must be between 0 and 1. The predicted probabilities therefore differ depending on the link function.

the same issue arises in that many activation functions exist and different functions will lead to different predictions.

2.7. Variance function uncertainty

The variance function models the uncertainty in y for given values of x. Another way to think of a variance function is that it models the spread of points around the mean prediction (Fig. 8). The standard approach assumes that uncertainty in y is constant (Fig. 8A). However, when the variance is not constant, a model for σ is required (Fig. 8B). Just like modelling μ as a function of x, we now need to model σ as a function of x. This function could be a simple function of one x variable or a full neural network for all x variables [44]. The latter option involves creating a second neural network for the variance, but this doubles the complexity of the model and the training time.

In Fig. 8B we do not use x directly, but model σ as a function of μ – in other words, the uncertainty in y is proportional to the predicted value of y. This allows for a simple mean function such as $f_\sigma = \sigma_0 + \sigma_1 \mu$, where σ_0 and σ_1 are parameters that control the relationship between σ and μ. Since variances must be positive values, a link function is needed to constrain f_σ, and the softplus function $l_\sigma = \log(1 + \exp(f_\sigma(\cdot))$ is used here. The result is shown in Fig. 8B, where the 95% shaded prediction region better matches the spread of the data compared with assuming a constant variance (Fig. 8A).

Instead of using a variance function, another option is to transform the outcome variable (e.g. log, square-root, or inverse), so that the uncertainty in y is constant. Alternatively, some distributions such as the Poisson and Bernoulli have a defined relationship between the variance the mean, which allows for non-constant variances, but they are only appropriate for certain type of data.

3. Sources of uncertainty combined

Breaking down the sources of uncertainty into seven items enables us to think about them separately and assess their importance when developing a prediction model. A final model may include several sources and they can be easily combined. For example, suppose variance and link functions were not required but two mean functions and two distribution functions performed similarly and therefore four models with each combination of distribution and mean function are fit to the training data and the predictions averaged. If fully Bayesian models are used, parameter uncertainty is already account for. And if the test data are measured with error, we can use the approach in 5 B to draw multiple samples for each test sample and feed them all through the prediction models. The more sources of uncertainty accounted for the more complex the prediction model. Hence, sources of uncertainty that make little contribution to the overall prediction uncertainty can be ignored.

4. Model fitting

A generic Bayesian 2-parameter exponential model is shown below, and each subsection above modified this basic model in different ways to highlight the sources of uncertainty.

$$\begin{aligned} y &\sim \text{Normal}(\mu, \sigma) && \text{Distribution function} \\ \mu &= \theta_2\,(1 - e^{-\theta_1 x}) && \text{Mean function} \\ \theta_1 &\sim \text{Truncated Normal}(1, 5) && \text{Priors} \\ \theta_2 &\sim \text{Truncated Normal}(0, 5) \\ \sigma &\sim \text{Truncated Normal}(0, 5) \end{aligned}$$

y is modelled a Normal distribution with a mean μ and standard deviation σ. The second line defines the mean function, and there is no variance or link function defined. The next three lines specify the prior distributions for the three parameters in the model. Given the mean function, we know that θ_1 and θ_2 are positive, and so is σ, by definition. Hence, we use truncated normal distributions with a lower limit of truncation at zero to represent our prior uncertainty in the parameters.

All the models were fit to the data with the Turing package in Julia. The No-U-Turn Sampler (NUTS) was used to update the uncertainty in the parameters after conditioning on the data. Three chains with 10,000 samples each were used and convergence was assessed with graphical and numeric summaries (trace plots and $\hat{R}$ statistics). Predictions were then generated for new values of x by (1) drawing samples from the updated parameter distributions, (2) using these to calculate μ, and (3) using μ and σ to generate values for y. Finally, any summary statistic can be calculated, such as the proportion of samples from y that are greater than a threshold value.

5. Uncertainty for classification tasks

The previous examples had a continuous outcome variable, but often outcomes are categorical such as toxic versus safe. Much of the previous discussion applies, but an important distinction is between uncertainty in a parameter (μ) and uncertainty in a prediction for a new observable (y) [5,17]. Greater parameter uncertainty leads to greater prediction uncertainty, but not for classification tasks, where the objective is to predict which of K classes a sample belongs to. This point is illustrated in Fig. 9.

Fig. 9 A plots data for a 2-group classification task with two predictors (x_1, x_2), and assume the grey triangles are the “toxic” class and the blue circles are the “safe” class. The black line is the optimal separating boundary. A logistic regression model is used to separate the classes and the prediction from the model will be a number between 0 and 1, where 1 corresponds toxic and zero corresponds to safe. This prediction is derived from the mean function and is passed through a link function to constrain the predictions to lie between 0 and 1. We’ll call these predicted values μ and the uncertainty in the prediction σ. Fig. 9B plots μ versus σ for each point in Fig. 9A, and the inverted-U relationship is a known feature of such models. But some samples are especially uncertain and are highlighted in red ($\sigma \geq 0.8$). Samples with intermediate uncertainty (σ between 0.6 and 0.8) are highlighted in orange. Fig. 9C shows that the uncertain samples are all close to the decision boundary, and that the most uncertain red points lie near the edge of the data where the location of decision boundary itself is uncertain. The shaded grey region in Fig. 9C represents the uncertainty in the decision boundary, and note how the uncertainty is wider at the ends compared with the middle.

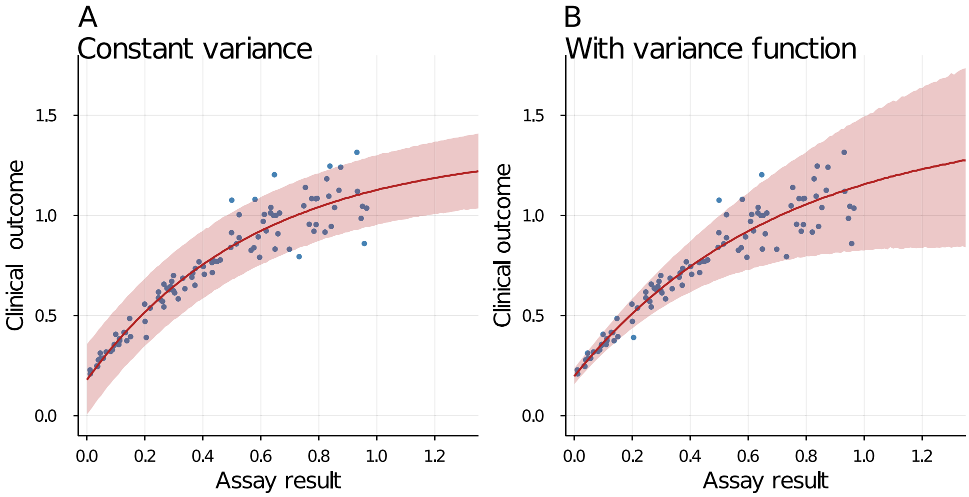

Fig. 8. Variance function uncertainty. A constant variance implies that the uncertainty in a prediction is the same for all values x (A). However, just like the mean prediction can change as function of x, so can the uncertainty in the prediction (B).

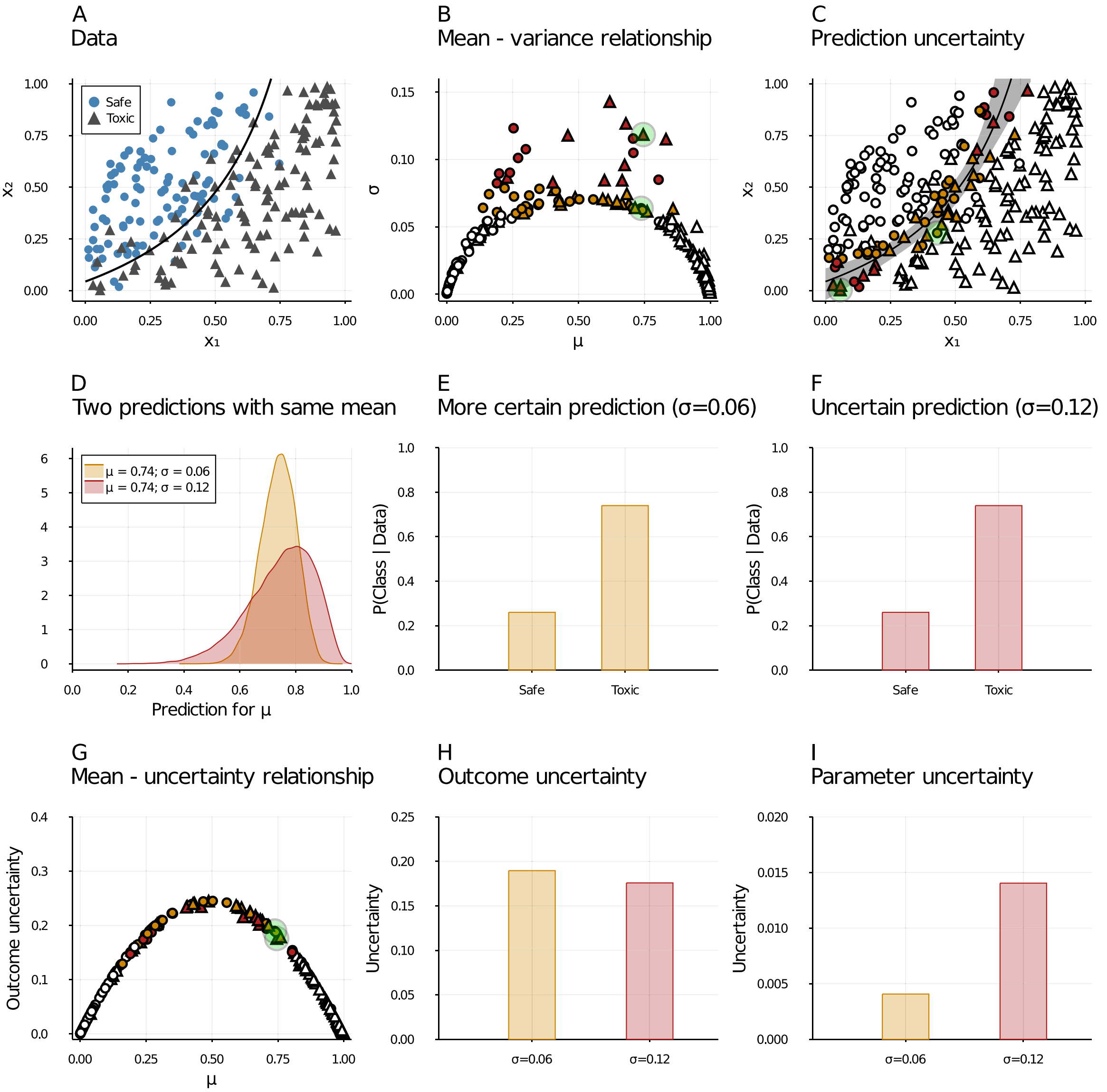

Fig. 9. Uncertainty in classification. Simulated data with two features (A). A plot of the mean (μ) and uncertainty (σ) shows the expected inverted-U relationship (B). The highest variance predictions (red points) are near the decision boundary and at the edge of the data, and high variance predictions (orange points) follow the boundary (C). Two compounds with the same mean but different uncertainties (D), have the same uncertainty in the final predicted value for y (E,F). Outcome uncertainty is largely explained by μ (G) and is similar for the two compounds (H). Parameter uncertainty is nearly 3.5 times greater for the compound with the larger σ (I). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 9 D plots the full distribution for two samples that have the same μ, but one has twice the uncertainty (these are highlighted with green circles in Fig. 9 B, C, and G). Recall that these distributions are for the parameter μ and not for the observable outcome y, which is either safe or toxic. To obtain a prediction for the observable we need a data generating distribution such as the Bernoulli distribution, giving us $y \sim \text{Bernoulli}(\mu)$. Fig. 9E and F shows the predicted values of y for these two compounds and note that they are identical, despite different values of σ. At first this may seem strange, and it occurs because the Bernoulli distribution doesn't have a σ parameter, and so this information is lost. Consider tossing a coin four times and getting 3 heads, our prediction for the probability of heads is $3/4 = 0.75$ (and $1/4 = 0.25$ for tails). Similarly, if we toss a coin 1000 times and get 750 heads, our prediction is still 0.75, even though we are much more certain that the proportion of heads is 0.75 (i.e. σ is much smaller). The width of the distributions in Fig. 9D (captured by σ) provides the *weight of evidence* [26] which quantifies how much the prediction will change as more data are gathered or as the inputs change. For example, a single new observation with four coin tosses alters the probability to either $3/5 = 0.6$ or $4/5 = 0.8$, depending on whether a heads or tails was observed, whereas with 1000 tosses the probability is still 0.75, rounding to two decimal places.

Uncertainty in the predicted classes is often divided into aleatoric and epistemic uncertainty [25,28]. Aleatoric uncertainty is supposedly due to “inherent randomness” whereas epistemic uncertainty is due to a lack of knowledge. Without starting a philosophical debate, we take the position that all uncertainty is due to a lack of knowledge [5,26]. Nevertheless, we can decompose our uncertainty into two components, which we call the *outcome uncertainty* and *parameter uncertainty*, and which correspond to aleatoric and epistemic uncertainty, respectively. The first component is how close μ is to zero or one – a more confident prediction would be close to these bounds, and the most uncertain prediction would be $\mu = 0.5$. This corresponds to outcome uncertainty. The second component is how confident we are in μ. When a weatherperson states there is a 70% chance of rain tomorrow, they do not mean exactly 70.00000% but 70% ± some amount. The uncertainty in the stated value corresponds to parameter uncertainty and is represented by the width of the distributions in Fig. 9D and the parameter σ (If the model included other sources of uncertainty, these would also be captured by σ and hence σ would represent more than just parameter uncertainty). Using the approach of Kwon et al. we decompose the two sources of uncertainty for the compounds and plot outcome uncertainty versus the mean prediction (μ) in Fig. 9G [30]. Note how μ largely explains the outcome uncertainty. The two compounds have similar outcome uncertainty since they have a similar value of μ (Fig. 9H). However, the compound with the larger value of σ has nearly 3.5 times greater parameter uncertainty (Fig. 9I). Parameter uncertainty or the weight of evidence (σ), provides important information about the uncertainty of a prediction, which is critical for high-stakes decisions.

6. Discussion

6.1. Generalisations and extensions

The above examples used simple models but this framework can be generalised to more complex cases. For example, we had functions for the mean and variance, but any parameter in the distribution function can be modelled. For example, a Student-t distribution has a parameter called the degrees of freedom (df) which controls the heaviness of the tails. The df could be modelled as a function of x, just like μ or σ [52].

The above examples used a single distribution function, but flexibility can be increased by using mixtures of distributions. For example, outliers can be modelled with a mixture of Gaussian distributions: one to account for the regular observations and the second to account for the outliers. Metabolite, gene, and protein levels are non-negative and often positively skewed, and hence gamma or lognormal distributions may be appropriate. But these distributions are only defined for values *greater* than zero, and there may be zeros in the data, which are often dealt with by adding a small value to all data points. A better option can be to model the data with a two-part model, one which accounts for the zeros and the other (e.g. gamma or lognormal) which accounts for the non-zero values. Such “hurdle models” provide this flexibility and also return a parameter that estimates the proportion of zeros, which may be scientifically interesting [13]. Taking this idea a step further, Dirichlet Process models allow us to specify as many distributions as needed to model the data. Instead of specifying a single distribution, we specify a prior over distributions, and learn them from the data (yes, we can specify a distribution over distributions! [42]).

The above examples also used a single mean function for each model, but it's possible to have a distribution of mean functions, which are called Gaussian Process models [19,49,54]. These flexible models can fit complex relationships between x and y. Surprisingly, they are not implemented via the mean function, but by generalising the variance function to make it a covariance function. Covariance functions are not discussed here but they are also useful for modelling hierarchical or nested data [24,48], and for modelling dependencies in time or space. Neural networks [21,56,68] and Bayesian additive regression trees (BART) [12,60] are other options for flexible mean functions.

6.2. Further advantages of PPMs

In addition to providing prediction uncertainty, PPMs have several other benefits. Hyperparameter values are typically selected by trying many options and choosing the combination that performs best. To avoid overfitting, crossvalidation or a similar approach divides the training data into smaller subsets, some of which are used for training and others to assess performance. But with small datasets, crossvalidation can give unstable models and a poor assessment of performance. Many Bayesian approaches can learn values of some hyperparameters using all the training data and have a built-in prevention of overfitting [64,71]. They also incorporate the uncertainty in the hyperparameters in the predictions. Models can still be compared using only the training data by estimating leave-one-out (LOO) crossvalidation performance, without the computational cost of actually retraining the model for each sample [66,67]. Vehtari and colleagues have also developed methods to assess when a LOO estimate is unreliable, and the model can be retrained only for these samples [69].

Another advantage is that background information such as adverse outcome pathways [7], constraints on parameters [34], or monotonic relationships [14] can often be incorporated into the model, which can guide the model to better solutions.

Often several structurally similar compounds are available that have different binding affinities or potencies, but also with different results in toxicity assays, and a decision must be taken to designate one compound in the series as the lead. PPMs can not only rank compounds but also obtain an uncertainty in the ranking, thus enabling decision makers to conclude that one compound is reliably better than another [34,55].

Finally, many popular machine learning methods have a PPM or Bayesian analogue, including regularised linear and generalised linear models (lasso, ridge regression) [10,47,53,45], tree models (random forests, xgboost) [11,12,60], support vector machines [59,64], and neural networks [43,56,68]. Hence, it is often possible to convert your favourite model into one that provides prediction uncertainty.

6.3. Drawbacks and challenges

The main drawback of Bayesian or other PMMs is that they require more work, possibly twice as much, since getting appropriately calibrated uncertainty is just as hard as getting accurate predictions. For example, 95% prediction intervals should contain 95% of the out-of-sample or test data values[72].

For fully Bayesian methods, the computational overhead may be high, making it difficult to iteratively fit, check, and update models during development (although computations are often much quicker when making predictions). Storage for parameter values may be a problem for large models since this equals the number of parameters times number of Markov chain Monte Carlo draws. These approaches may therefore be harder to scale to large datasets, but faster and scalable algorithms is an active area of research. Another solution to large data is to cleverly select a weighted subset of samples that is much smaller than the original but captures the essential features. This "coreset" approach enables standard PPM methods to be used on the smaller dataset with little loss of information [8,22].

Finally, not all sources of uncertainty can be captured. Many sources of uncertainty discussed above arise because many modelling options are available, and different choices lead to different predictions. All of the choices relate to the prediction model, but many decisions need to be made outside of the model. We refer to these extra-model choices as the project workflow and they include experimental decisions such as the technology, cell-line, assay, antibodies, protocol, and so on. Also included are data processing pipelines where raw data are cleaned, transformed, categorised, coded, and normalised before they are entered into a prediction model. A single workflow is commonly used, with the untested assumption that variations in the workflow will lead to the same predictions and results. However, variations in workflows and analytic decisions do lead to variations results [4,23,32,57,58,61,62].

6.4. Reporting uncertainty to help risk communication and decision making

The ultimate aim of prediction models in drug discovery is to enable better decision making. Thus, not only should predictions be accurate with prediction uncertainty adequately represented, but the results should be easy to understand by decision makers. Fortunately, PPMs provide intuitive results for continuous (Fig. 6D), binary (Fig. 9D), categorical, and ordered categorical outcomes [56,71], as well as for compound rankings [34,55]. We have found that safety pharmacologists and other project members can easily interpret the predictive distributions provided by PPMs and value the confidence in the predictions that these distributions provide [34,71].

With recent advances in algorithms, hardware, and software, Bayesian or other PPMs are now feasible for most – if not all – machine learning problems encountered in drug discovery. Making PPMs the standard approach for critical ML problems will enable more informed and better decisions.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Bezanson J, Edelman A, Karpinski S, Shah VB. Julia: a fresh approach to numerical computing. SIAM Rev 2017;59(1):65–98.

[2] Blackwell M, Honaker J, King G. A unified approach to measurement error and missing data: overview and applications. Sociol Methods Res 2015;46(3):303–41.

[3] Blundell C, Cornebise J, Kavukcuoglu K, Wierstra D. Weight uncertainty in neural networks. In: Proceedings of the 32nd international conference on international conference on machine learning. In: ICML'15, 37; 2015. p. 1613–22.

[4] Botvinik-Nezer R, Holzmeister F, Camerer CF, Dreber A, Huber J, Johannesson M, Kirchler M, Iwanir R, Mumford JA, Adcock RA, Avesani P, Baczkowski BM, Bajracharya A, Bakst L, Ball S, Barilari M, Bault N, Beaton D, Beitner J, Benoit RG, Berkers RMWJ, Bhanji JP, Biswal BB, Bobadilla-Suarez S, Bortolini T, Bottenhorn KL, Bowring A, Braem S, Brooks HR, Brudner EG, Calderon CB, Camilleri JA, Castrellon JJ, Cecchetti L, Cieslik EC, Cole ZJ, Collignon O, Cox RW, Cunningham WA, Czoschke S, Dadi K, Davis CP, Luca AD, Delgado MR, Demetriou L, Dennison JB, Di X, Dickie EW, Dobryakova E, Donnat CL, Dukart J, Duncan NW, Durnez J, Eed A, Eickhoff SB, Erhart A, Fontanesi L, Fricke GM, Fu S, Galván A, Gau R, Genon S, Glatard T, Glerean E, Goeman JJ, Golowin SAE, González-García C, Gorgolewski KJ, Grady CL, Green MA, Guassi Moreira JF, Guest O, Hakimi S, Hamilton JP, Hancock R, Handjaras G, Harry BB, Hawco C, Herholz P, Herman G, Heunis S, Hoffstaedter F, Hogeveen J, Holmes S, Hu CP, Huettel SA, Hughes ME, Iacovella V, Iordan AD, Isager PM, Isik AI, Jahn A, Johnson MR, Johnstone T, Joseph MJE, Juliano AC, Kable JW, Kassinopoulos M, Koba C, Kong XZ, Koscik TR, Kucukboyaci NE, Kuhl BA, Kupek S, Laird AR, Lamm C, Langner R, Lauharatanahirun N, Lee H, Lee S, Leemans A, Leo A, Lesage E, Li F, Li MYC, Lim PC, Lintz EN, Liphardt SW, Vermeer AB, Losecaat Love BC, Mack ML, Malpica N, Marins T, Maumet C, McDonald K, McGuire JT, Melero H, Méndez Leal AS, Meyer B, Meyer KN, Mihai G, Mitsis GD, Moll J, Nielson DM, Nilsonne G, Notter MP, Olivetti E, Onicas AI, Papale P, Patil KR, Peelle JE, Pérez A, Pischedda D, Poline JB, Prystauka Y, Ray S, Reuter-Lorenz PA, Reynolds RC, Ricciardi E, Rieck JR, Rodriguez-Thompson AM, Romyn A, Salo T, Samanez-Larkin GR, Morales E, Schlichting ML, Schultz DH, Shen Q, Sheridan MA, Silvers JA, Skagerlund K, Smith A, Smith DV, Sokol-Hessner P, Steinkamp SR, Tashjian SM, Thirion B, Thorp JN, Tinghög G, Tisdall L, Tompson SH, Toro-Serey C, Torre Tresols JJ, Tozzi L, Truong V, Turella L, van 't Veer AE, Verguts T, Vettel JM, Vijayarajah S, Vo K, Wall MB, Weeda WD, Weis S, White DJ, Wisniewski D, Xifra-Porxas A, Yearling EA, Yoon S, Yuan R, Yuen KSL, Zhang L, Zhang X, Zosky JE, Nichols TE, Poldrack RA, Schonberg T. Variability in the analysis of a single neuroimaging dataset by many teams. Nature 2020;582:84–8.

[5] Briggs W. Uncertainty: the soul of modeling, probability and statistics. New York, NY: Springer; 2016.

[6] Brown T.B., Mann B., Ryder N., Subbiah M., Kaplan J., Dhariwal P., Neelakantan A., Shyam P., Sastry G., Askell A., Agarwal S., Herbert-Voss A., Krueger G., Henighan T., Child R., Ramesh A., Ziegler D.M., Wu J., Winter C., Hesse C., Chen M., Sigler E., Litwin M., Gray S., Chess B., Clark J., Berner C., McCandlish S., Radford A., Sutskever I., Amodei D.. Language models are few-shot learners. 2020. ArXiv.

[7] Burgoon LD, Angrish M, Garcia-Reyero N, Pollesch N, Zupanic A, Perkins E. Predicting the probability that a chemical causes steatosis using adverse outcome pathway bayesian networks (AOPBNs). Risk Anal 2019;40(3):512–23.

[8] Campbell T, Broderick T. Bayesian coreset construction via greedy iterative geodesic ascent. In: Dy J, Krause A, editors. Proceedings of the 35th international conference on machine learning. Proceedings of Machine Learning Research, 80. PMLR; 2018. p. 698–706.

[9] Carroll RJ, Ruppert D, Stefanski LA, Crainiceanu CM. Measurement error in nonlinear models: a modern perspective. 2nd ed. Boca Raton, FL: Chapman & Hall/CRC; 2006.

[10] Carvalho CM, Polson NG, Scott JG. Handling sparsity via the horseshoe. Proc Mach Learn Res 2009;5:73–80.

[11] Chipman HA, George EI, McCulloch RE. Bayesian CART model search. J Am Stat Assoc 1998;93(443):935–48.

[12] Chipman HA, George EI, McCulloch RE. BART: Bayesian additive regression trees. Ann Appl Stat 2010;4(1):266–98.

[13] Cragg JG. Some statistical models for limited dependent variables with application to the demand for durable goods. Econometrica 1971;39(5):829.

[14] DePalma G, Craig BA. Bayesian monotonic errors-in-variables models with applications to pathogen susceptibility testing. Stat Med 2017;37(3):487–502.

[15] Gal Y, Ghahramani Z. Dropout as a bayesian approximation: representing model uncertainty in deep learning. In: Balcan MF, Weinberger KQ, editors. Proceedings of The 33rd International Conference on Machine Learning. Proceedings of Machine Learning Research, 48. New York, New York, USA: PMLR; 2016. p. 1050–9.

[16] Ge H, Xu K, Ghahramani Z. Turing: a language for flexible probabilistic inference. In: International conference on artificial intelligence and statistics, AISTATS 2018, 9–11 April 2018, Playa Blanca, Lanzarote, Canary Islands, Spain; 2018. p. 1682–90.

[17] Geisser S. Predictive inference: an introduction. New York, NY: Chapman & Hall; 1993.

[18] Gelman A, Carlin JB, Stern HS, Rubin DB. Bayesian data analysis. 2nd ed. Boca Raton: Chapman & Hall/CRC; 2004.

[19] Gramacy RB. Surrogates: gaussian process modeling, design, and optimization for the applied sciences. CRC Press; 2020.

[20] Gustafson P. Measurement error and misclassification in statistics and epidemiology: impacts and bayesian adjustments. Boca Raton: Chapman & Hall/CRC; 2004.

[21] Hirschfeld L, Swanson K, Yang K, Barzilay R, Coley CW. Uncertainty quantification using neural networks for molecular property prediction. J Chem Inf Model 2020;60(8):3770–80.

[22] Huggins J.H., Campbell T., Broderick T.. Coresets for scalable Bayesian logistic regression. 2016. ArXiv.

[23] Huntington-Klein N, Arenas A, Beam E, Bertoni M, Bloem JR, Burli P, et al. The influence of hidden researcher decisions in applied microeconomics. Econ Inq 2021.

[24] Johnstone RH, Bardenet R, Gavaghan DJ, Mirams GR. Hierarchical Bayesian inference for ion channel screening dose-response data. Wellcome Open Res 2016;1:6.

[25] Kendall A, Gal Y. What uncertainties do we need in Bayesian deep learning for computer vision?. Advances in Neural Information Processing Systems, 30. Curran Associates, Inc.; 2017.

[26] Keynes JM. A treatise on probability. London: Macmillan & Co; 1921.

[27] Khosravi A, Nahavandi S, Creighton D, Atiya AF. Comprehensive review of neural network-based prediction intervals and new advances. IEEE Trans Neural Netw 2011;22(9):1341–56.

[28] Kiureghian AD, Ditlevsen O. Aleatory or epistemic? Does it matter? Struct Saf 2009;31(2):105–12.

[29] Kristiadi A, Hein M, Hennig P. Being Bayesian, even just a bit, fixes overconfidence in ReLU networks. In: I HDII, Singh A, editors. Proceedings of the 37th international conference on machine learning. Proceedings of Machine Learning Research, 119. PMLR; 2020. p. 5436–46.

[30] Kwon Y, Won JH, Kim BJ, Paik MC. Uncertainty quantification using Bayesian neural networks in classification: application to biomedical image segmentation. Comput Stat Data Anal 2020;142:106816.

[31] Lakshminarayanan B, Pritzel A, Blundell C. Simple and scalable predictive uncertainty estimation using deep ensembles. In: Advances in neural information processing systems, 30; 2017. p. 6402–13.

[32] Landy JF, Jia ML, Ding IL, Viganola D, Tierney W, Dreber A, et al. Crowdsourcing hypothesis tests: making transparent how design choices shape research results. Psychol Bull 2020;146(5):451–79.

[33] Lazic SE. Four simple ways to increase power without increasing the sample size. Lab Anim 2018;52(6):621–9.

[34] Lazic SE, Edmunds N, Pollard CE. Predicting drug safety and communicating risk: benefits of a bayesian approach. Toxicol Sci 2018;162(1):89–98.

[35] Lazic SE, Williams DP. Improving drug safety predictions by reducing poor analytical practices. Toxicol Res Appl 2020;4 239784732097863.

[36] Lesaffre E, Lawson AB. Bayesian biostatistics. Chichester, UK: Wiley; 2012.

[37] Little RJA, Rubin DB. Statistical analysis with missing data. 3rd ed. Hoboken, NJ: WIley; 2020.

[38] Lunn D, Jackson C, Best N, Thomas A, Spiegelhalter D. The BUGS book: a practical introduction to bayesian analysis. Boca Raton, FL: CRC Press; 2013.

[39] McElreath R. Statistical rethinking: Bayesian course with examples in R and Stan. Boca Raton, FL: CRC Press; 2016.

[40] Mervin LH, Johansson S, Semenova E, Giblin KA, Engkvist O. Uncertainty quantification in drug design. Drug Discov Today 2021;26(2):474–89.

[41] Muff S, Riebler A, Held L, Rue H, Saner P. Bayesian analysis of measurement error models using integrated nested Laplace approximations. J R Stat Soc 2014;64(2):231–52.

[42] Muller P, Quintana FA, Jara A, Hanson T. Bayesian nonparametric data analysis. Springer; 2015.

[43] Neal RM. Bayesian learning for neural networks. New York, NY: Springer; 1996.

[44] Nix D, Weigend A. Estimating the mean and variance of the target probability distribution. In: Proceedings of 1994 IEEE international conference on neural networks (ICNN'94). IEEE; 1994.

[45] Park T, Casella G. The Bayesian lasso. J Am Stat Assoc 2008;103(482):681–6.

[46] Pearce T, Leibfried F, Brintrup A. Uncertainty in neural networks: approximately Bayesian ensembling. In: Chiappa S, Calandra R, editors. Proceedings of the twenty third international conference on artificial intelligence and statistics. Proceedings of Machine Learning Research, 108. PMLR; 2020. p. 234–44.

[47] Piironen J, Vehtari A. Sparsity information and regularization in the horseshoe and other shrinkage priors. Electron J Stat 2017;11(2):5018–51.

[48] Pinheiro JC, Bates DM. Mixed-effects models in S and S-Plus. London: Springer; 2000.

[49] Rasmussen CE, Williams CKI. Gaussian processes for machine learning. Cambridge, MA: MIT Press; 2006.

[50] Reynolds J, Malcomber S, White A. A Bayesian approach for inferring global points of departure from transcriptomics data. Computat Toxicol 2020 100138.

[51] Richardson S, Gilks WR. A Bayesian approach to measurement error problems in epidemiology using conditional independence models. Am J Epidemiol 1993;138(6):430–42.

[52] Rigby R. Distributions for modelling location, scale, and shape : using GAMLSS in R. Bocat Raton, FL: CRC Press; 2020.

[53] Ročková V, George EI. The spike-and-slab LASSO. J Am Stat Assoc 2018;113(521):431–44.

[54] Schulz E, Speekenbrink M, Krause A. A tutorial on gaussian process regression: modelling, exploring, and exploiting functions. J Math Psychol 2018;85:1–16.

[55] Semenova E., Guerriero M.L., Zhang B., Hock A., Hopcroft P., Kadamur G., Afzal A.M., Lazic S.E.. Flexible fitting of PROTAC concentration-response curves with Gaussian processes. 2020a. BioRxiv.

[56] Semenova E, Williams DP, Afzal AM, Lazic SE. A Bayesian neural network for toxicity prediction. Computat Toxicol 2020;16:100133.

[57] Shi L, Campbell G, Jones WD, Campagne F, Wen Z, Walker SJ, et al. The microarray quality control (MAQC)-II study of common practices for the development and validation of microarray-based predictive models. Nat Biotechnol 2010;28:827–38.

[58] Silberzahn R, Uhlmann EL, Martin DP, Anselmi P, Aust F, Awtrey E, et al. Many analysts, one data set: making transparent how variations in analytic choices affect results. Adv Methods Pract Psychol Sci 2018;1(3):337–56.

[59] Sollich P. Bayesian methods for support vector machines: evidence and predictive class probabilities. Mach Learn 2002;46(1/3):21–52.

[60] Sparapani RA, Logan BR, McCulloch RE, Laud PW. Nonparametric survival analysis using Bayesian Additive Regression Trees (BART). Stat Med 2016;35(16):2741–53.

[61] Stanton-Geddes J, de Freitas CG, Dambros CdS. In defense of P values: comment on the statistical methods actually used by ecologists. Ecology 2014;95:637–42.

[62] Steegen S, Tuerlinckx F, Gelman A, Vanpaemel W. Increasing transparency through a multiverse analysis. Perspect Psychol Sci 2016;11(5):702–12.

[63] Teye M, Azizpour H, Smith K. Bayesian uncertainty estimation for batch normalized deep networks. In: Dy J, Krause A, editors. Proceedings of the 35th international conference on machine learning. Proceedings of Machine Learning Research, 80. PMLR; 2018. p. 4907–16.

[64] Tipping ME. Sparse bayesian learning and the relevance vector machine. JMLR 2001;1(1):211–44.

[65] van Buuren S. Flexible imputation of missing data. Boca Raton, FL: CRC Press; 2012.

[66] Vehtari A, Gelman A, Gabry J. Erratum to: practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. Stat Comput 2016;27(5) 1433–1433.

[67] Vehtari A, Gelman A, Gabry J. Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. Stat Comput 2016;27(5):1413–32.

[68] Vehtari A, Lampinen J. Bayesian neural networks: case studies in industrial applications. In: Soft Computing in Industrial Applications. Springer London; 2000. p. 415–24.

[69] Vehtari A., Simpson D., Gelman A., Yao Y., Gabry J.. Pareto smoothed importance sampling. 2015. ArXiv 1507.02646.

[70] Welling M, Teh YW. Bayesian learning via stochastic gradient Langevin dynamics. In: Proceedings of the 28th international conference on international conference on machine learning. In: ICML'11, 33; 2011. p. 681–8.

[71] Williams DP, Lazic SE, Foster AJ, Semenova E, Morgan P. Predicting drug-induced liver injury with Bayesian machine learning. Chem Res Toxicol 2020;33(1):239–48.

[72] Zhang Y, Lee AA. Bayesian semi-supervised learning for uncertainty-calibrated prediction of molecular properties and active learning. Chem Sci 2019;10(35):8154–63.