Proper scoring rules

Finn Lindgren

January 29, 2019

1 Predictive distributions

We are interested in estimating a parameter vector $\boldsymbol{\theta} \in \mathbb{R}^p$, given observed values $\mathcal{Y}_{\text{obs}} = \{y_i, i = 1, \dots, n\}$, for some likelihood model $L(\boldsymbol{\theta}; \mathcal{Y}_{\text{obs}})$, and then use the estimated model to *forecast* or *predict* the values of some *test data* $\mathcal{Y}_{\text{test}}$. By producing not only point predictions but also quantifying the prediction uncertainty due to inherent randomness and parameter estimation error uncertainty, e.g. in the form of a full predictive distribution or just a prediction variance, we can then compare different models and estimation methods in terms of how good their assessment *scores* are.

Likelihood theory can be used to show that, for large observation samples, the maximum likelihood estimation error is approximately Normal,

$$\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}_{\text{true}} \sim \mathrm{N}(\mathbf{0}, \boldsymbol{\Sigma}_\theta) \tag{1}$$

where $\boldsymbol{\Sigma}_\theta^{-1} = H(\widehat{\boldsymbol{\theta}})$, the Hessian of the negative log-likelihood, is a plug-in estimate of the error covariance matrix. To simplify the notation for tracking the parameter estimation uncertainty, we will treat the unknown parameter vector $\boldsymbol{\theta}$ as a random vector which is approximately $\mathrm{N}(\widehat{\boldsymbol{\theta}}, \boldsymbol{\Sigma}_\theta)$. This way, once we have found $\widehat{\theta}$ and $\boldsymbol{\Sigma}_\theta$, we only need the conditional distributions for the observations, given the parameter values.

Note: In classical *frequentist statistics*, this is a slight abuse of notation, since the true $\boldsymbol{\theta}$ value, $\boldsymbol{\theta}_{\text{true}}$, is not random in that setting. However, in *Bayesian statistics* this is a more directly valid approach, given an improper uniform density prior for $\boldsymbol{\theta}$.

Given the distribution properties of an test observation y given $\boldsymbol{\theta}$, e.g. as a probability density $f_{y|\boldsymbol{\theta}}(\cdot)$, we can write the full predictive/forecast density as

$$f_y(y) = \int_{\mathbb{R}} f_{y|\boldsymbol{\theta}}(y) f_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \,\mathrm{d}\boldsymbol{\theta}, \tag{2}$$

where $f_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ is the density of the parameter error uncertainty model, $\mathrm{N}(\widehat{\boldsymbol{\theta}}, \boldsymbol{\Sigma}_\theta)$. In general, this integral might be difficult to evaluate, but there are special cases where the answer is known.

We will in general identify the notation F with both the abstract *predictive distribution* and the concrete *predictive cumulative distribution function*, $F(x) = \mathrm{P}(y \leq x) = \int_{-\infty}^x f_y(y) \,\mathrm{d}y$.

1.1 A Gaussian example with partly known parameters

Assume that, for some known p-vector $\boldsymbol{z}$, the observations follow the random model $y = \boldsymbol{z}^\top \boldsymbol{\theta} + \epsilon$, where $\epsilon \in \mathrm{N}(0, \sigma_\epsilon^2)$ for some known value σ_ϵ. Then $(y|\boldsymbol{\theta}) \sim \mathrm{N}(\boldsymbol{z}^\top \boldsymbol{\theta}, \sigma_\epsilon^2)$, and according to our approximate uncertainty assumption about $\boldsymbol{\theta}$, we get

$$y \sim \mathrm{N}(\boldsymbol{z}^\top \widehat{\boldsymbol{\theta}}, \boldsymbol{z}^\top \boldsymbol{\Sigma}_\theta \boldsymbol{z} + \sigma_\epsilon^2). \tag{3}$$

1.2 General notation and predictive moments

Working with the full predictive distribution is often difficult, so we here focus on Normal/Gaussian predictive distributions, or predictions that only involve the 1st and 2nd order moments of the predictive distribution.

Using the expectation *tower property*, $\mathrm{E}_A(A) = \mathrm{E}_A[\mathrm{E}_{A|B}(A)]$, we can write

$$\begin{aligned} \mu_F &= \mathrm{E}_F(y) = \mathrm{E}_\theta[\mathrm{E}_{y|\theta}(y)] && (4) \\ \sigma_F^2 &= \mathrm{Var}_F(y) = \mathrm{E}_\theta[\mathrm{Var}_{y|\theta}(y)] + \mathrm{Var}_\theta[\mathrm{E}_{y|\theta}(y)] && (5) \end{aligned}$$

where the second line is the tower property for variances, that can be derived from $\mathrm{Var}(A) = \mathrm{E}\{[A - \mathrm{E}(A)]^2\}$. These identities often provide practical solutions to finding the forcast/prediction mean and standard deviations, μ_F and σ_F.

1.3 Example: non-constant variace

Generalising the model from Lab 3, we define a model where $(y|\boldsymbol{\theta})$ are independent and Normal/Gaussian, and the expectation and log-variance are linear in $\boldsymbol{\theta}$,

$$\begin{aligned} \mathrm{E}_{y|\theta}(y) &= \boldsymbol{z}_E^\top \boldsymbol{\theta} && (6) \\ \log[\mathrm{Var}_{y|\theta}(y)] &= \boldsymbol{z}_V^\top \boldsymbol{\theta} && (7) \end{aligned}$$

The $\boldsymbol{z}_E$ and $\boldsymbol{z}_V$ vectors can be stacked as rows of model matrices $\boldsymbol{Z}_E$ and $\boldsymbol{Z}_V$, so that the expectation of a vector of observations can be written $\boldsymbol{Z}_E\boldsymbol{\theta}$.

Combining the conditional moments for $(y|\boldsymbol{\theta})$ with the uncertainty model for $\boldsymbol{\theta}$, we obtain

$$\begin{aligned} \mathrm{E}_F(y) &= \boldsymbol{z}_E^\top \widehat{\boldsymbol{\theta}} && (8) \\ \mathrm{Var}_F(y) &= \mathrm{E}_\theta\left[\exp\left(\boldsymbol{Z}_V\boldsymbol{\theta}\right)\right] + \mathrm{Var}_\theta\left(\boldsymbol{z}_E^\top\boldsymbol{\theta}\right) && (9) \end{aligned}$$

The second term of the variance is

$$\begin{aligned} \mathrm{Var}_\theta\left(\boldsymbol{z}_E^\top\boldsymbol{\theta}\right) &= \mathrm{Cov}_\theta\left(\boldsymbol{z}_E^\top\boldsymbol{\theta}, \boldsymbol{z}_E^\top\boldsymbol{\theta}\right) && (10) \\ &= \boldsymbol{z}_E^\top \mathrm{Cov}_\theta\left(\boldsymbol{\theta}, \boldsymbol{\theta}\right)\boldsymbol{z}_E && (11) \\ &= \boldsymbol{z}_E^\top \boldsymbol{\Sigma}_\theta \boldsymbol{z}_E. && (12) \end{aligned}$$

The first term of the variance is more difficult. We will use the known result (either from the *log-Normal distribution* or the *moment generating function* for the Normal distribution) that if $x \sim \mathrm{N}(\mu, \sigma^2)$, then $\mathrm{E}(\mathrm{e}^x) = \mathrm{e}^{\mu+\sigma^2/2}$:

$$\begin{aligned} \boldsymbol{z}_V^\top\boldsymbol{\theta} &\sim \mathrm{N}(\boldsymbol{z}_V^\top\widehat{\boldsymbol{\theta}}, \boldsymbol{z}_V^\top\boldsymbol{\Sigma}_\theta\boldsymbol{z}_V) && (13) \\ \mathrm{E}_\theta[\exp(\boldsymbol{z}_V^\top\boldsymbol{\theta})] &= \exp\left(\boldsymbol{z}_V^\top\widehat{\boldsymbol{\theta}} + \boldsymbol{z}_V^\top\boldsymbol{\Sigma}_\theta\boldsymbol{z}_V/2\right). && (14) \end{aligned}$$

Combining the results, we get the predictive variance as

$$\sigma_V^2 = \mathrm{Var}_F(y) = \exp\left(\boldsymbol{z}_V^\top\widehat{\boldsymbol{\theta}} + \boldsymbol{z}_V^\top\boldsymbol{\Sigma}_\theta\boldsymbol{z}_V/2\right) + \boldsymbol{z}_E^\top\boldsymbol{\Sigma}_\theta\boldsymbol{z}_E. \qquad (15)$$

1.3.1 Example model definition

We implement a simple version of the general model, by requiring each θ_i parameter to be used for either the expectation or the log-variance, but not both. We also use a single covariate, in the code called `x`, so that every observation is a pair (x_i, y_i), and $\mathrm{E}_{y|\theta}(y_i) = \theta_1 + x_i\theta_2$ and $\log[\mathrm{Var}_{y|\theta}(y_i)] = \theta_3 + x_i\theta_4$.

First, define a function that constructs the $\boldsymbol{Z}$ matrices:

```r
model_Z <- function(x) {
  Z0 <- model.matrix(~ 1 + x)
  list(ZE = cbind(Z0, Z0 * 0), ZV = cbind(Z0 * 0, Z0))
}
```

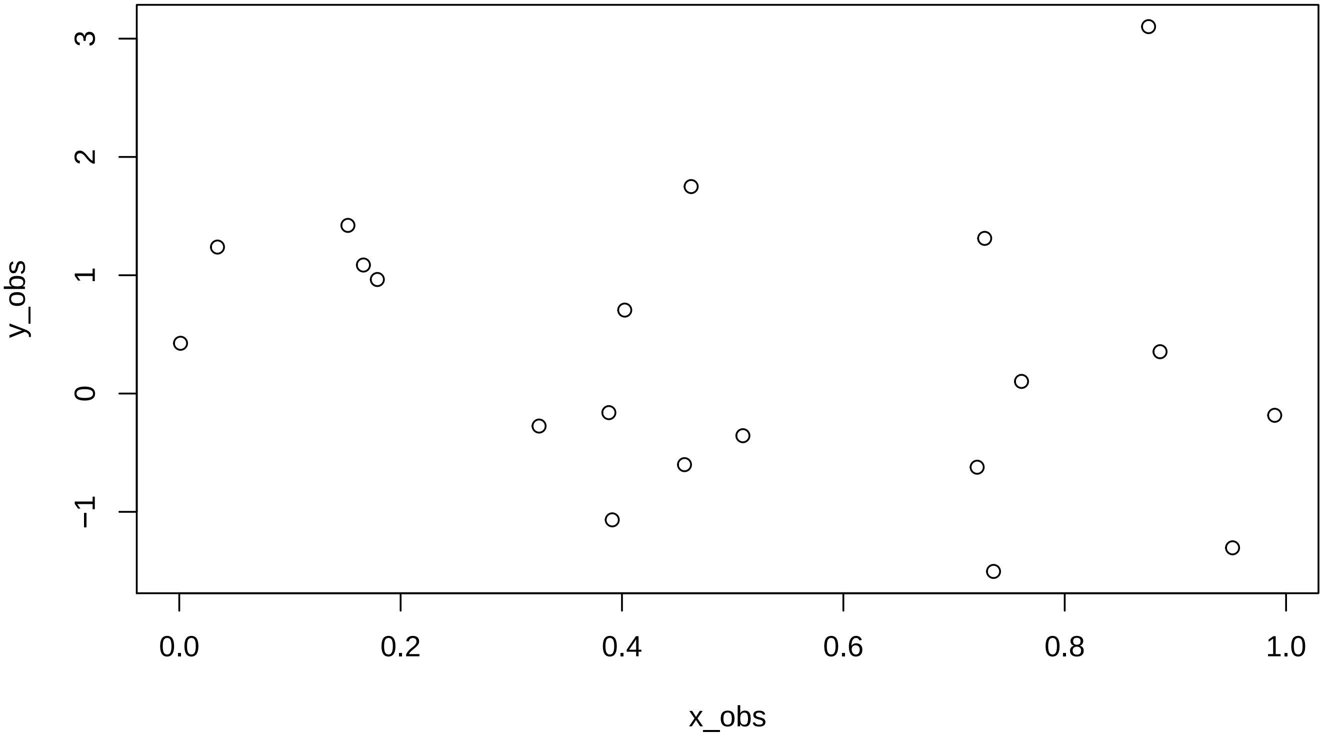
We will write the rest of the code so that we essentially only need to change the definition of `model_Z()` to run a different model of the same general class.

Then, a function that implements the general version of the negative log-likelihood, using a list of the type generated by `model_Z()` to known what the model is.

```r
neg_log_lik <- function(theta, Z, y) {
  -sum(dnorm(y, mean = Z$ZE %*% theta, sd = exp(Z$ZV %*% theta)^0.5, log = TRUE))
}
```

In order to have something to test, we generate a synthetic data sample:

```r
n <- 20
x_obs <- runif(n)
Z_obs <- model_Z(x_obs)
theta_true <- c(1, -2, -2, 4)
y_obs <- rnorm(n = n, mean = Z_obs$ZE %*% theta_true, sd = exp(Z_obs$ZV %*% theta_true)^0.5)
plot(x_obs, y_obs)
```



Treating the simulated data as our observed sample, we estimate the paramter vector $\boldsymbol{\theta}$ using `optim()`:

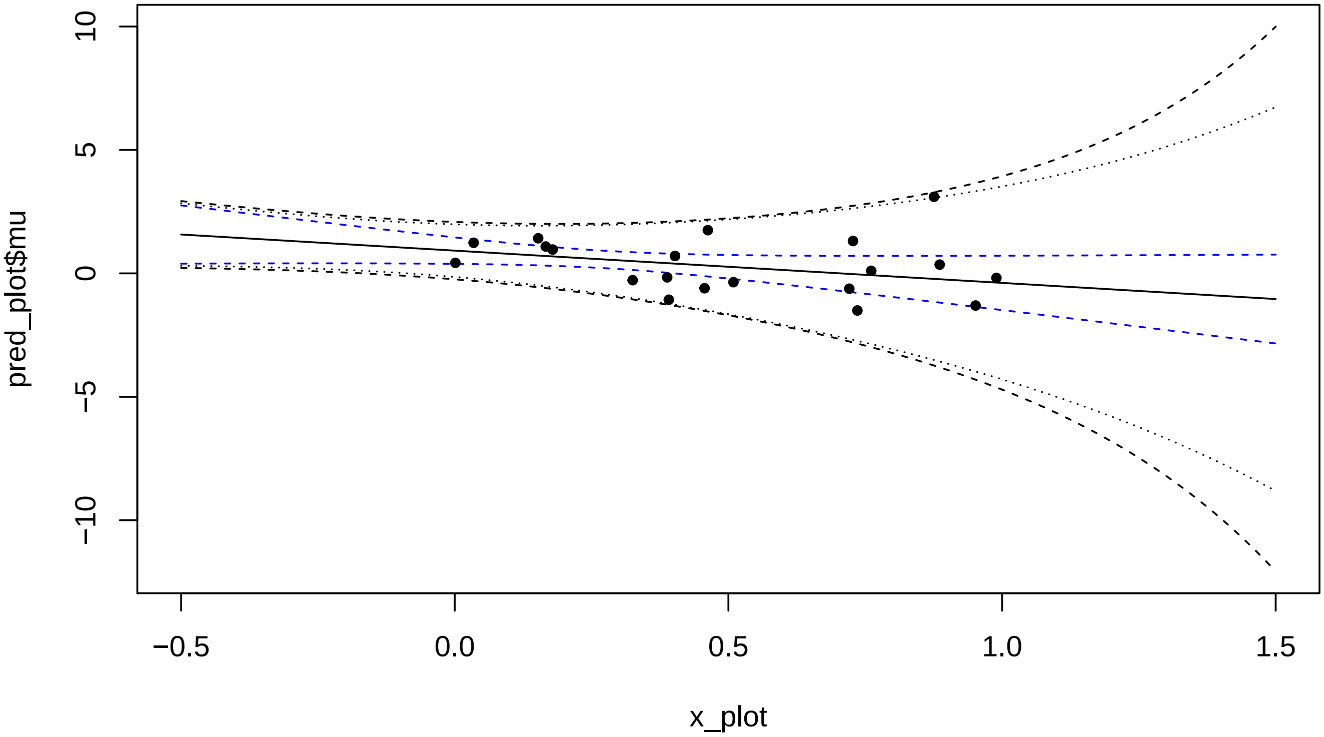
```
opt <- optim(rep(0, 4), fn = neg_log_lik, Z = Z_obs, y = y_obs,
             method = "BFGS", hessian = TRUE)
theta_hat <- opt$par
Sigma_theta <- solve(opt$hessian)
```

Next, we define a function with behaviour similar to `predict()`, that we'll use to compute predictive distributions and prediction intervals:

```
# Value: data.frame with columns (mu, sigma, lwr, upr)
model_predict <- function(theta, data, Sigma_theta = NULL,
                          type = c("expectation", "log-variance", "observation"),
                          alpha = 0.05, df = Inf,
                          nonlinear.correction = TRUE) {
  type <- match.arg(type)
  Z <- model_Z(data) ## Note: Will use model_Z() defined in the global workspace!
  fit_E <- Z$ZE %*% theta
  fit_V <- Z$ZV %*% theta
  if (is.null(Sigma_theta)) {
    ZE_var <- 0
    ZV_var <- 0
  } else {
    ZE_var <- rowSums(Z$ZE * (Z$ZE %*% Sigma_theta))
    ZV_var <- rowSums(Z$ZV * (Z$ZV %*% Sigma_theta))
  }
  if (type == "expectation") {
    fit <- fit_E
    sigma <- ZE_var^0.5
  } else if (type == "log-variance") {
    fit <- fit_V
    sigma <- ZV_var^0.5
  } else if (type == "observation") { ## observation predictions
    fit <- fit_E
    sigma <- (exp(fit_V + ZV_var / 2 * nonlinear.correction) + ZE_var)^0.5
  }
  q <- qt(1 - alpha / 2, df = df)
  lwr <- fit - q * sigma
  upr <- fit + q * sigma
  data.frame(mu = fit, sigma, lwr, upr)
}
```

Now, let's plot the estimates and predictions!

```
x_plot <- seq(-0.5, 1.5, length=100)
conf_plot <- model_predict(theta_hat, x_plot, Sigma_theta = Sigma_theta, type = "expectation")
pred_plot <- model_predict(theta_hat, x_plot, Sigma_theta = Sigma_theta, type = "observation")
pred_plot2 <- model_predict(theta_hat, x_plot, Sigma_theta = Sigma_theta, type = "observation",
                            nonlinear.correction = FALSE)
plot(x_plot, pred_plot$mu, type = "l", ylim = range(pred_plot[, c("lwr", "upr")]))
lines(x_plot, pred_plot$lwr, lty = 2)
lines(x_plot, pred_plot$upr, lty = 2)
lines(x_plot, pred_plot2$lwr, lty = 3)
lines(x_plot, pred_plot2$upr, lty = 3)
lines(x_plot, conf_plot$lwr, lty = 2, col=4)
lines(x_plot, conf_plot$upr, lty = 2, col=4)
points(x_obs, y_obs, pch=20)
```



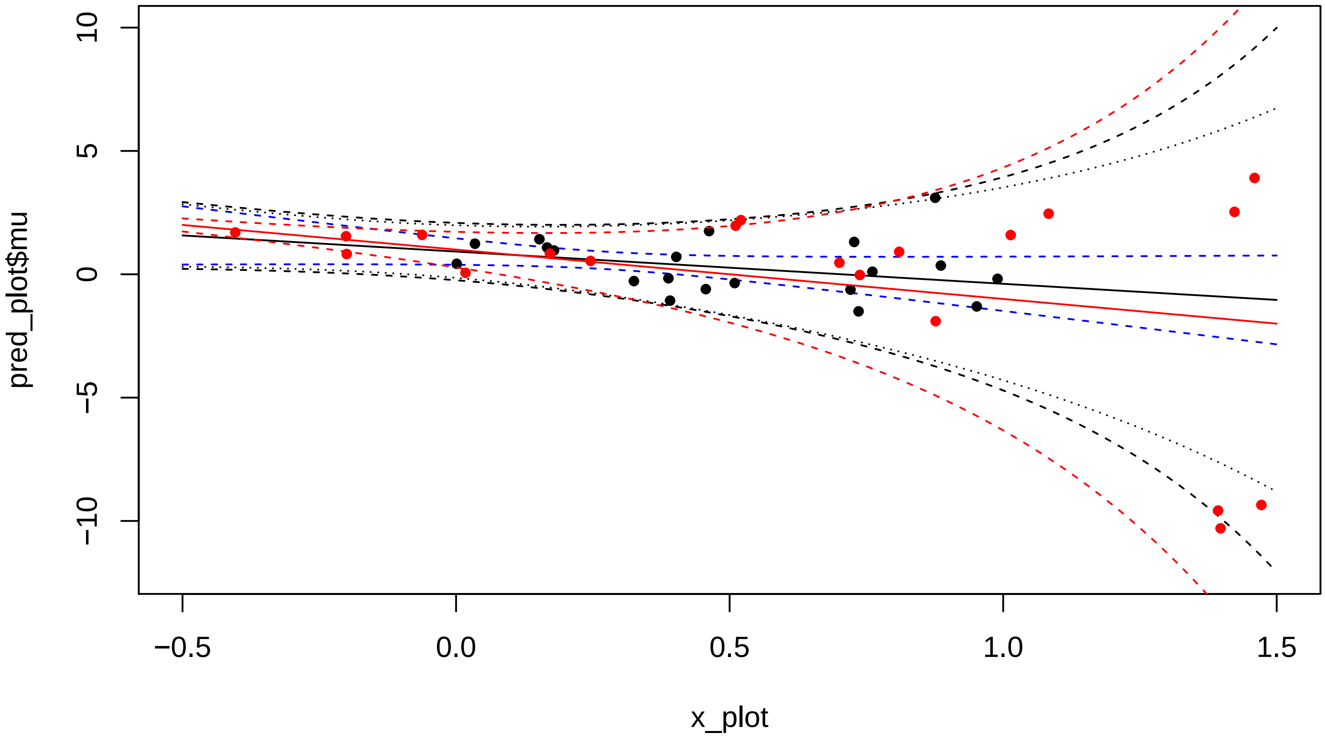
1.3.2 Towards model assessment with test data

We want to assess how good the estimated model is at predicting unseen data; i.e. data that wasn't used when estimating the model parameters. We simulate some new test data from the true model (in real data, this would have been a held-out part of the raw observed data):

```
x_test <- runif(20, -0.5, 1.5)
Z_test <- model_Z(x_test)
y_test <- rnorm(n = length(x_test),
                mean = Z_test$ZE %*% theta_true,
                sd = exp(Z_test$ZV %*% theta_true)^0.5)
```

Let's do a visual inspection:

```
plot(x_plot, pred_plot$mu, type = "l", ylim = range(pred_plot[, c("lwr", "upr")], y_test))
lines(x_plot, pred_plot$lwr, lty = 2)
lines(x_plot, pred_plot$upr, lty = 2)
lines(x_plot, pred_plot2$lwr, lty = 3)
lines(x_plot, pred_plot2$upr, lty = 3)
lines(x_plot, conf_plot$lwr, lty = 2, col=4)
lines(x_plot, conf_plot$upr, lty = 2, col=4)
points(x_obs, y_obs, pch=20)
points(x_test, y_test, pch=20, col=2)
true_plot <- model_predict(theta_true, x_plot, type = "observation")
lines(x_plot, true_plot$mu, lty = 1, col = 2)
lines(x_plot, true_plot$lwr, lty = 2, col = 2)
lines(x_plot, true_plot$upr, lty = 2, col = 2)
```



We see that the estimated model (black) is close to the true model (red), and that the nonlinear correction term from the exponential expectation gives an important contribution (dotted vs dashed black curves).

The next step is to introduce more formal and quantifiable assessment techniques in the form of *proper scoring rules*.

2 Scoring rules

We want to assess how *far away* from the truth our forecast/prediction distributions are. To do this, we might consider a number of quantities that measure the discrepancy in different ways:

- Squared Error (SE):

  $$S_{\text{SE}}(F, y) = (y - \widehat{y}_F)^2 \tag{16}$$

  where $\widehat{y}_F$ is a point estimate under F, e.g. the expectation μ_F.

- Absolute Error (AE):

  $$S_{\text{AE}}(F, y) = |y - \widehat{y}_F| \tag{17}$$

  where $\widehat{y}_F$ is a point estimate under F, e.g. the predictive median $F^{-1}(1/2)$.

- Logarithmic/Ignorance score (LOG/IGN):

  $$S_{\text{LOG}}(F, y) = -\log f(y) \tag{18}$$

  where $f(\cdot)$ is the predictive density.

- Dawid-Sebastiani score (DS):

  $$S_{\text{DS}}(F, y) = \frac{(y - \mu_F)^2}{\sigma_F^2} + \log(\sigma_F^2) \tag{19}$$

  Note: If F is Normal, then $S_{\text{DS}}(F, y) = 2S_{\text{LOG}}(F, y) - \log(2\pi)$

- Continuous Ranked Probability Score (CRPS):

  $$S_{\text{CRPS}}(F, y) = \int_{\mathbb{R}} [F(x) - \mathbb{I}(y \leq x)]^2 \, \mathrm{d}x \tag{20}$$

  This can be seen as a generalisation of AE that also cares about other quantiles than the median.

These scores are defined to be *negatively oriented*, meaning that the *lower the score, the better*.

2.1 Defining scores in R

Thinking back at the output from the example `model_predict()`, we can define R functions that mimic the $S(F, y)$ notation. For SE and DS we define `score_se()` and `score_ds()`:

```r
# Input:
#   pred : data.frame with (at least) a column "mu"
#   y : data vector
score_se <- function(pred, y) {
  (y - pred$mu)^2
}
# Input:
```

```
#   pred : data.frame with (at least) columns "mu" and "sigma"
#   y : data vector
score_ds <- function(pred, y) {
  ((y - pred$mu) / pred$sigma)^2 / 2 + log(pred$sigma)
}
```

2.2 Evaluating scores

We can now evaluate the scores for the example. We include the scores for a simplistic model that assumes that all the observations have a common expectation and variance, $(y_i|\boldsymbol{\theta}) \sim \mathrm{N}(\beta_0, \sigma_\epsilon^2)$ (for brevity, we ignore some of the parameter uncertainty when constructing the prediction from this model, in `pred0`).

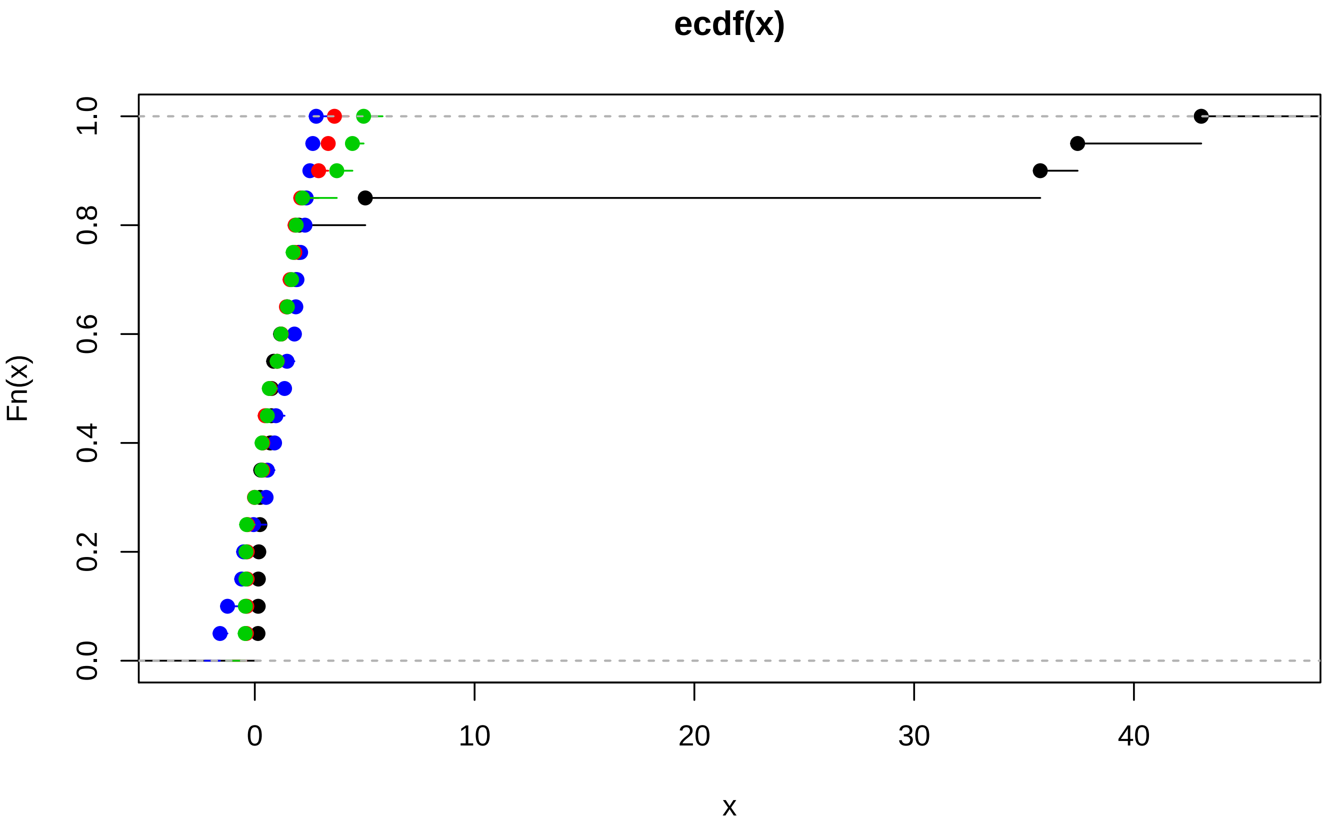
```
opred_true <- model_predict(theta_true, x_obs, type = "observation")
opred0 <- data.frame(mu = mean(y_obs), sigma = sd(y_obs))
opred1 <- model_predict(theta_hat, x_obs, Sigma_theta = Sigma_theta, type = "observation")
opred2 <- model_predict(theta_hat, x_obs, Sigma_theta = Sigma_theta, type = "observation",
                        nonlinear.correction = FALSE)
pred_true <- model_predict(theta_true, x_test, type = "observation")
pred0 <- data.frame(mu = mean(y_obs), sigma = sd(y_obs))
pred1 <- model_predict(theta_hat, x_test, Sigma_theta = Sigma_theta, type = "observation")
pred2 <- model_predict(theta_hat, x_test, Sigma_theta = Sigma_theta, type = "observation",
                       nonlinear.correction = FALSE)
## SE for observed and test data
rbind(
  c(mean(score_se(opred_true, y_obs)),
    mean(score_se(opred0, y_obs)),
    mean(score_se(opred1, y_obs)),
    mean(score_se(opred2, y_obs))),
  c(mean(score_se(pred_true, y_test)),
    mean(score_se(pred0, y_test)),
    mean(score_se(pred1, y_test)),
    mean(score_se(pred2, y_test)))
)

##           [,1]      [,2]      [,3]      [,4]
## [1,]  1.429616  1.247714  1.215824  1.215824
## [2,] 13.800697 17.275713 14.711117 14.711117

## DS for observed and test data
rbind(
  c(mean(score_ds(opred_true, y_obs)),
    mean(score_ds(opred0, y_obs)),
    mean(score_ds(opred1, y_obs)),
    mean(score_ds(opred2, y_obs))),
  c(mean(score_ds(pred_true, y_test)),
    mean(score_ds(pred0, y_test)),
    mean(score_ds(pred1, y_test)),
    mean(score_ds(pred2, y_test)))
)
```

```
##           [,1]     [,2]      [,3]      [,4]
## [1,] 0.5011834 0.611303 0.4714792 0.4603268
## [2,] 1.1011282 6.713104 1.0495769 1.2065578
```

We see that the SE appears to be less sensitive to model mis-specification than DS. We also see that the scores for the true model are generally *worse* than for the estimated model when applied to the observed data. This is because the estimated models are adapted to the random values that happened to be observed. For the test data, at least the mis-specified model is clearly worse than the other estimated models as well as the true model. In our example we see that the estimated model predictions `pred1` score better at DS than the true model even for the test data, but with a different random seed for the pseudo-random numbers, this might very well be reversed. In general, one wants to use many observations for parameter estimation, but at the same time have enough test observations for reliable model assesment.

We can also plot the individual scores, to get an overview of the contributions to the average scores:

```
plot.ecdf(score_ds(pred0, y_test))
plot.ecdf(score_ds(pred_true, y_test), add=TRUE, col=4) # blue
plot.ecdf(score_ds(pred1, y_test), add=TRUE, col=2)     # red
plot.ecdf(score_ds(pred2, y_test), add=TRUE, col=3)     # green
```



3 Score expectations and proper scoring rules

What functions of the predictive distributions are useful scores? We want to reward accurate (unbiased) and precise (small variance) predictions, but not at the expense of understating true uncertainty.

First, we define the expectation of a score as

$$S(F,G) = \mathrm{E}_{y\sim G}[S(F,y)] \tag{21}$$

A negatively oriented score is *proper* if it fulfils

$$S(F,G) \geq S(G,G). \tag{22}$$

The practical interpretation of this is that a proper score does not reward cheating; stating a lower (or higher) forecast/prediction uncertainty will not, on average, give a better score than stating the truth.

A proper score that has equality of the expectations *only* when F and G are the same, $F(\cdot) \equiv G(\cdot)$, is said to be *strictly proper*.

Let's revisit some of our previously defined scores and check if they are proper!

3.1 SE

$$\begin{aligned}
S_{\mathrm{SE}}(F,G) &= \mathrm{E}_{y\sim G}[S_{\mathrm{SE}}(F,y)] && (23)\\
&= \mathrm{E}_{y\sim G}[(y-\mu_F)^2] && (24)\\
&= \mathrm{E}_{y\sim G}[(y-\mu_G+\mu_G-\mu_F)^2] && (25)\\
&= \mathrm{E}_{y\sim G}[(y-\mu_G)^2 + 2(y-\mu_G)(\mu_G-\mu_F) + (\mu_G-\mu_F)^2] && (26)\\
&= \mathrm{E}_{y\sim G}[(y-\mu_G)^2] + 2(\mu_G-\mu_F)\mathrm{E}_{y\sim G}[y-\mu_G] + (\mu_G-\mu_F)^2 && (27)\\
&= \sigma_G^2 + (\mu_G-\mu_F)^2 && (28)
\end{aligned}$$

This is minimised when $\mu_F = \mu_G$. Therefore $S_{\mathrm{SE}}(F,G) \geq S_{\mathrm{SE}}(G,G) = \sigma_G^2$, so the score is proper. It is not strictly proper, since there are many different distributions with expectation μ_G.

3.2 DS

$$\begin{aligned}
S_{\mathrm{DS}}(F,G) &= \mathrm{E}_{y\sim G}[S_{\mathrm{DS}}(F,y)] && (29)\\
&= \frac{\mathrm{E}_{y\sim G}[(y-\mu_F)^2]}{\sigma_F^2} + \log(\sigma_F^2) && (30)\\
&= \frac{\sigma_G^2 + (\mu_G-\mu_F)^2}{\sigma_F^2} + \log(\sigma_F^2) && (31)
\end{aligned}$$

This is minimised when $\mu_F = \mu_G$ and $\sigma_F = \sigma_G$. Therefore $S_{\mathrm{DS}}(F,G) \geq S_{\mathrm{DS}}(G,G) = 1 + \log(\sigma_G^2)$, so the score is proper. It is not strictly proper, since there are many different distributions with expectation μ_G and standard deviation σ_G.

3.3 CRPS

By definition, we have a relationship between the cdf $G(\cdot)$ and the expectation of an indicator function:

$$\mathrm{E}_{y\sim G}[\mathbb{I}(y \leq x)] = \mathrm{P}(y \leq x) = G(x). \tag{32}$$

By expanding the square in the integrand for the CRPS definition, and changing order between expectation and integration, we can rewrite the CRPS expectation as follows:

$$\begin{aligned}
S_{\text{CRPS}}(F, G) &= \mathrm{E}_{y\sim G}\left\{\int_{\mathbb{R}} [F(x) - \mathbb{I}(y \leq x)]^2 \,\mathrm{d}x\right\} && (33)\\
&= \mathrm{E}_{y\sim G}\left\{\int_{\mathbb{R}} \left[F(x)^2 - 2F(x)\mathbb{I}(y \leq x) + \mathbb{I}(y \leq x)\right] \,\mathrm{d}x\right\} && (34)\\
&= \int_{\mathbb{R}} \left[F(x)^2 - 2F(x)G(x) + G(x)\right] \,\mathrm{d}x && (35)\\
&= \int_{\mathbb{R}} \left[F(x)^2 - 2F(x)G(x) + G(x)^2 + G(x) - G(x)^2\right] \,\mathrm{d}x && (36)\\
&= \int_{\mathbb{R}} [F(x) - G(x)]^2 \,\mathrm{d}x + \int_{\mathbb{R}} G(x)\,[1 - G(x)] \,\mathrm{d}x && (37)
\end{aligned}$$

This is minimised when $F(\cdot) \equiv G(\cdot)$, so the score is proper. Furthermore, $S_{\text{CRPS}}(F, G) = S_{\text{CRPS}}(G, G)$ *only* when $F(\cdot) \equiv G(\cdot)$, so the score is strictly proper.

4 Score distributions

When we take the average of n predictive scores for a model, we should take the random variability into account when comparing with scores for other models. We'll take a brief look at these score distributions, in particular $\mathrm{E}_{y\sim F}[S(F, y)]$ and $\mathrm{Var}_{y\sim F}[S(F, y)]$. When taking average scores, a full analysis should take into account that the prediction are typically *dependent*, leading to larger average score variances, but we'll ignore that here.

We define *average* or *mean* scores under two scenarios:

1. A single forcast/prediction distribution F, with many independent observations y_i
2. A collection of forcast/prediction distributions $\{F_i\}$, each predicting a single observation from the collection $\{y_i\}$, i.e. we have a collection of prediction/obserrvation pairs $\{(F_i, y_i)\}$.

For case 1, the average score is

$$\overline{S}(F, \{y_i\}) = \frac{1}{n}\sum_{i=1}^{n} S(F, y_i). \tag{38}$$

For case 2, the average score is

$$\overline{S}(\{(F_i, y_i)\}) = \frac{1}{n}\sum_{i=1}^{n} S(F_i, y_i). \tag{39}$$

For example, the commonly used *Mean Squared Error* (MSE) can be defined as MSE $= \overline{S}_{\text{SE}}(\{(F_i, y_i)\})$. This also means that MSE is a proper scoring rule.

4.1 SE

We already saw that $\mathrm{E}_{y\sim F}[S_{\text{SE}}(F,y)] = \sigma_F^2$. The *variance* of the score depends on the type of the predictive distribution. Since our predictive distributions are usually close to Normal, we consider that assumption to get a useful general approximation:

$$\begin{aligned}
\mathrm{Var}_{y\sim F}[S_{\text{SE}}(F,y)] &= \mathrm{E}_{y\sim F}\left\{[(y-\mu_F)^2-\sigma_F^2]^2\right\} && (40)\\
&= \mathrm{E}_{y\sim F}\left[(y-\mu_F)^4-2\sigma_F^2(y-\mu_F)^2+\sigma_F^4\right] && (41)\\
&= 3\sigma_F^4-2\sigma_F^4+\sigma_F^4 && (42)\\
&= 2\sigma_F^4. && (43)
\end{aligned}$$

Furthermore, we know that a rescaled version of the square has a $\chi^2(1)$ distribution: $\{S_{\text{SE}}(F,y)/\sigma_F^2 | y\sim F\} \sim \chi^2(1)$.

For the average score under a common F, we get $\overline{S}_{\text{SE}}(F,\{y_i\}) \sim \frac{\sigma_F^2}{n}\chi^2(n)$, which has expectation σ_F^2 and variance $2\sigma_F^4/n$.

For the average score under different F_i, the resulting weighted sum of χ^2-distributions doesn't have a simple form, but we can get the expectation and variance:

$$\mathrm{E}_{\{y_i\sim F_i\}}\left[\overline{S}_{\text{SE}}(\{(F_i,y_i)\})\right] = \frac{1}{n}\sum_{i=1}^n \sigma_{F_i}^2 \tag{44}$$

$$\mathrm{Var}_{\{y_i\sim F_i\}}\left[\overline{S}_{\text{SE}}(\{(F_i,y_i)\})\right] = \frac{2}{n^2}\sum_{i=1}^n \sigma_{F_i}^4 \tag{45}$$

4.2 DS

Since the DS score is closely related to the SE score, the analysis follow the same principles. If F is Normal, each normalised fraction $(y-\mu_F)/\sigma_F$ in the DS definition is $\mathrm{N}(0,1)$, so the dependence on i in the score only enters through the offset term $\log(\sigma_F^2)$:

$$\{S_{\text{DS}}(F,y)|y\sim F\} \sim \chi^2(1)+\log(\sigma_F^2) \tag{46}$$

$$\{\overline{S}_{\text{DS}}(F,\{y_i\})|\{y_i\sim F\}\} \sim \frac{1}{n}\chi^2(n)+\log(\sigma_F^2) \tag{47}$$

$$\{\overline{S}_{\text{DS}}(\{(F_i,y_i)\})|\{y_i\sim F_i\}\} \sim \frac{1}{n}\chi^2(n)+\frac{1}{n}\sum_{i=1}^n\log(\sigma_{F_i}^2) \tag{48}$$

$$\mathrm{E}_{\{y_i\sim F_i\}}\left[\overline{S}_{\text{DS}}(\{(F_i,y_i)\})\right] = 1+\frac{1}{n}\sum_{i=1}^n\log(\sigma_{F_i}^2) \tag{49}$$

$$\mathrm{Var}_{\{y_i\sim F_i\}}\left[\overline{S}_{\text{DS}}(\{(F_i,y_i)\})\right] = \frac{1}{n} \tag{50}$$

4.3 CRPS

The distribution of the CRPS is much harder to analyse. The expectation was derived earlier, and we only state the end result for the variance, as a double integral:

$$\mathrm{Var}_{y\sim F}\left[S_{\text{CRPS}}(F,y)\right] = \iint_{\mathbb{R}\times\mathbb{R}}[1-2F(u)][1-2F(v)]F[\min(u,v)]\{1-F[\max(u,v)]\}\,\mathrm{d}u\,\mathrm{d}v. \tag{51}$$