

Getting Started with SHELF

Jeremy Oakley

2019-03-26

1 Introduction

We illustrate the process of fitting a probability distribution to an expert's judgements, and obtaining feedback.

Various elicitation methods are suggested in SHELF, but they all involve either asking for probability or quantile judgements: the expert is either asked questions of the form, 'What is your probability $P(X \leq 10)$?' or, 'Provide a value x such that you think $P(X \leq x) = 0.5$.' Either way, we think of the expert as providing points on his/her cumulative distribution function (CDF).

As an example, suppose the uncertain quantity is the percentage X of voters who will vote for a particular candidate in an upcoming leadership election.

2 Eliciting individual distributions from multiple experts

Suppose we have two experts, who first consider their judgements independently. Expert A states

$$P(X \leq 25) = 0.25, P(X \leq 30) = 0.5, P(X \leq 35) = 0.75,$$

and Expert B states

$$P(X \leq 30) = 0.25, P(X \leq 35) = 0.5, P(X < 50) = 0.75.$$

The values of the quantiles are arranged in a matrix, with one column per expert

```
v <- matrix(c(25, 30, 35, 30, 35, 50), nrow = 3, ncol = 2)
```

```
##      [,1] [,2]
## [1,]  25  30
## [2,]  30  35
## [3,]  35  50
```

The probabilities should be arranged in matrix if they are different for the two experts (for example, if one expert had specified tertiles instead of lower and upper quartiles), otherwise they are stored in a vector

```
p <- c(0.25, 0.5, 0.75)
```

We now fit probability distributions to these judgements with the `fitdist` command. Without specifying any limits (`lower` and `upper`), only normal and Student- t distributions (with 3 degrees of freedom used as a default) will be fitted. Including a `lower` limit will result in log-normal, log Student- t and gamma distributions also being fitted, and including both a `lower` and `upper` limit will result in a beta distribution fit, scaled to the interval [`lower`, `upper`].

```
library(SHELF)
myfit <- fitdist(vals = v, probs = p, lower = 0, upper = 100)
```

The object `myfit` includes parameters of the fitted distributions for each expert, sums of squared errors (elicited - fitted probabilities) for each distribution and expert, and the original elicited judgements:

```
names(myfit)

## [1] "Normal"          "Student.t"       "Gamma"           "Log.normal"
## [5] "Log.Student.t"   "Beta"            "ssq"             "best.fitting"
## [9] "vals"            "probs"           "limits"
```

For example, the parameters of the fitted beta distributions are

```
myfit$Beta

##           shape1    shape2
## expert.A 11.58800 26.711428
## expert.B  3.76511  6.000839
```

Inspecting `myfit$ssq` (sum of squared differences between elicited and fitted probabilities), we see that the normal distribution fits best for Expert 1, and the log Student- t_3 distribution fits best for Expert 2 (although the fit would probably be judged adequate for any the distributions, in this case). These best-fitting distributions are used as defaults in the `plotfit` function, and the `feedback` function when used with multiple experts

```
myfit$ssq

##           Normal      Student-t      Gamma      Log normal      Log Student-t
## expert.A 1.109336e-31 5.564207e-12 0.000126733 0.0002833689 0.0002740339
## expert.B 9.495880e-03 9.351947e-03 0.007194602 0.0061172216 0.0059898413
##           Beta
## expert.A 4.194213e-05
## expert.B 8.591167e-03
```

We plot the fitted distributions, including a linear pool.

```
plotfit(myfit, lp = TRUE)
```

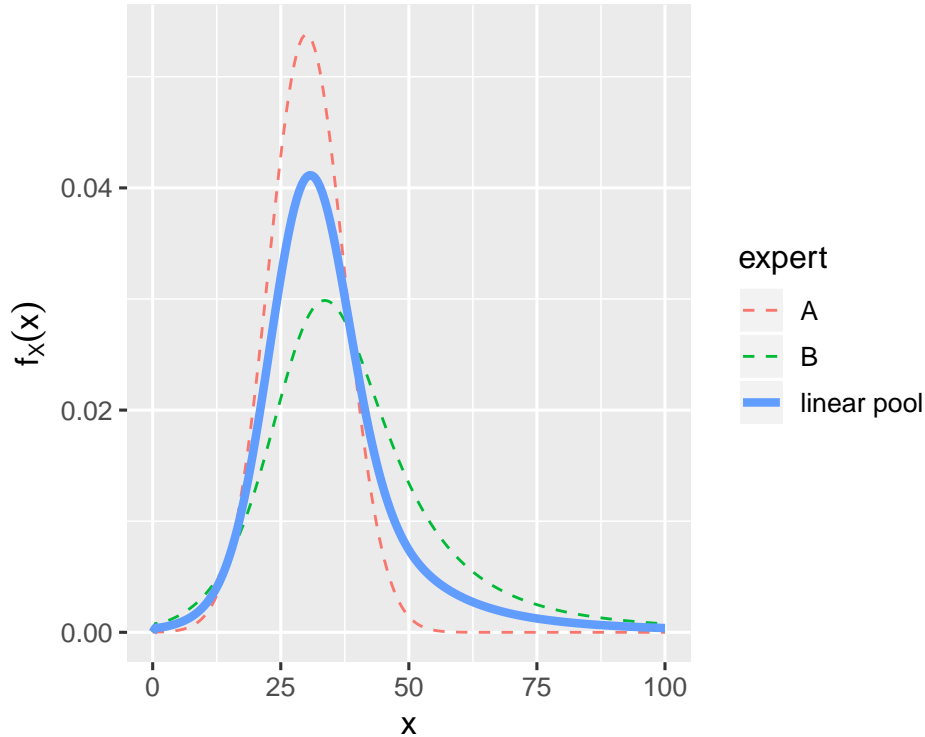


Figure 1: The two fitted distributions (normal for Expert 1 and log Student- t_3 for Expert 2) and an equal-weighted linear pool.

3 Eliciting a single distribution from an individual or a group of experts

Now we elicit a single a single ‘consensus’ distribution from the two experts. Suppose they agree

$$P(X \leq 25) = 0.25, P(X \leq 30) = 0.5, P(X \leq 40) = 0.75.$$

We fit a single set of distributions to these judgements.

```
v <-c(25, 30, 40)
p <-c(0.25, 0.5, 0.75)
consensus <- fitdist(vals = v, probs = p, lower = 0, upper = 100)
```

The best fitting distribution is the log Student- t_3 distribution, but the (scaled) beta distribution would likely be adequate at this stage, and we may choose to use the beta instead, given it has the appropriate bounds.

We plot the fitted density, and mark the fitted 5th and 95th percentiles

```
plotfit(consensus, ql = 0.05, qu = 0.95, d = "beta")
```

We can obtain values of fitted percentiles and probabilities with the `feedback` function. This will show fitted values for all the fitted distributions

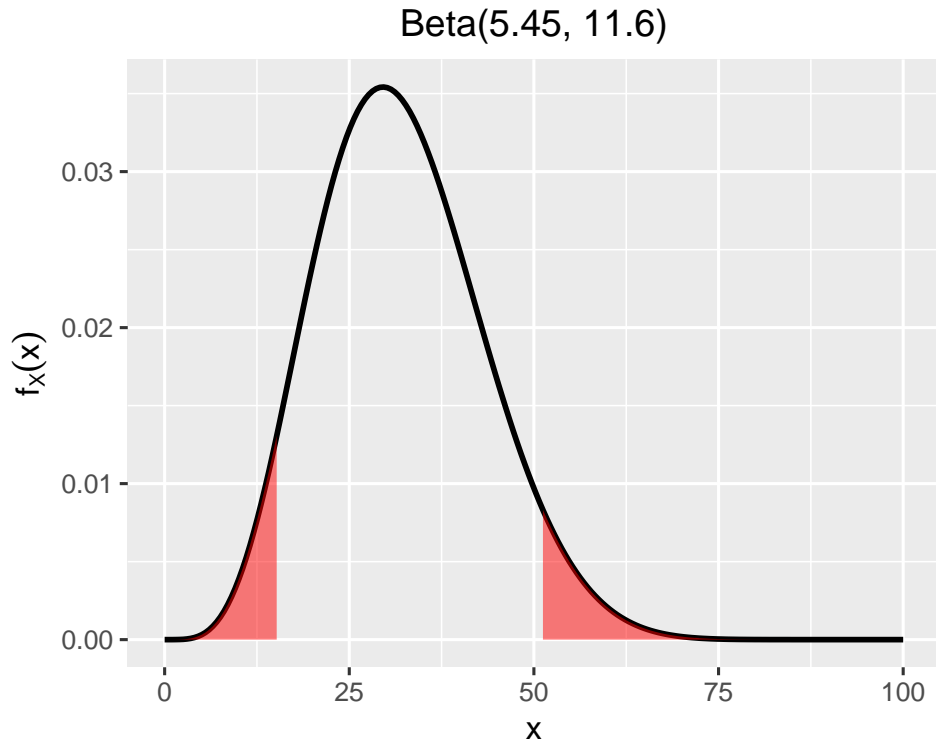


Figure 2: The fitted consensus distribution, with the lower and upper 5% tail areas shown as feedback.

```
feedback(consensus, quantiles = c(0.05, 0.95))
```

```
## $fitted.quantiles
##      Normal Student-t Gamma Log normal Log Student-t Beta Histogram
## 0.05  12.5      7.49 16.2      17.3      14.8 15.2      5
## 0.95  50.4     55.10 53.2     55.3     64.1 51.2     88
##
## $fitted.probabilities
##      elicited Normal Student-t Gamma Log normal Log Student-t Beta
## 25      0.25 0.288      0.289 0.279      0.274      0.275 0.283
## 30      0.50 0.451      0.453 0.461      0.466      0.469 0.456
## 40      0.75 0.772      0.774 0.769      0.767      0.768 0.770
##      Histogram
## 25      0.25
## 30      0.50
## 40      0.75
```

For the fitted beta distribution, we have $P(X < 15.2) = 0.05$, and we can also compare, for example, the fitted probability $P(X < 25) = 0.283$ with elicited probability $P(X < 25) = 0.25$.