Universal Probability Density Function of Detected SNR

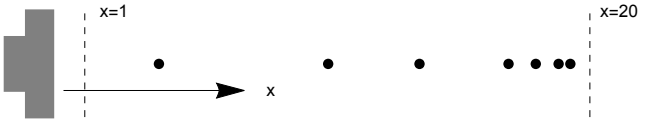
Marshall Bradley

Radar echoes measured on a signal-to-noise ratio scale that are greater than a threshold are shown to follow a Pareto probability distribution. The decay parameter of this distribution is directly related to the spreading loss mechanism coupling the radar transmitter to the target. The problem is motivated as an example of a selection effect in Bayesian probability theory. Numerical examples are presented using synthetic data and data from the WiPPR radar system.

Selection effects

Sometimes the data from an instrument is curated by the recording process in such a fashion that the full range of the data is not available. This could be a property of the instrument. It could also arise from the necessity to to record only those data values with sufficiently high signal-to-noise ratio in order to guarantee data integrity. These are both examples of selection effects. David MacKay in his book *Information Theory, Inference and Learning Algorithms* posed the problem by considering a question from an old Cambridge exam:

> "Unstable particles are emitted from a source and decay at a distance x, a real number that has an exponential probability distribution with characteristic length λ. Decay events can be observed only if they occur in a window extending from $x = 1$ cm to $x = 20$ cm. N decays are observed at locations $\{x_1, x_2 \ldots x_N\}$. What is λ?"



The question actually is somewhat more precisely, what is a reasonable way of estimating λ. Reasonableness in this case leads to a Bayesian analysis. This requires finding an expression for the likelihood of the data given a knowledge of the unknown parameter λ.

The likelihood of measuring a data value x given λ and the effect of measuring data only in the range $1 < x < 20$ is

$$L(x \mid \lambda) = \frac{1}{Z(\lambda)} \frac{1}{\lambda} \exp(-x/\lambda)$$

where

$$Z(\lambda) = \int_1^{20} \frac{1}{\lambda} \exp(-x/\lambda)\, d\lambda = \exp(-x) - \exp(-x/20)\,.$$

Note that $\lambda^{-1} \exp(-x/\lambda)$ is just the probability density function of the exponential distribution that generates the data. The likelihood of observing the data values $\{x_1, x_2 \ldots x_N\}$ is

$$L(x_1\, x_2 \ldots x_N \mid \lambda) = \frac{1}{(Z(\lambda)\,\lambda)^N} \exp\{-(x_1 + x_2 \ldots x_N)/\lambda\} = \frac{1}{(Z(\lambda)\,\lambda)^N} \exp(-N\bar{x}/\lambda)$$

where $\bar{x}$ is the mean of the data sample. Bayes theorem tells us that if the probability density function $f(\lambda)$ represents our prior knowledge about the parameter λ, then the corresponding posterior probability density function is

$$P(\lambda \mid x_1\, x_2 \ldots x_N) \propto \frac{1}{(Z(\lambda)\,\lambda)^N} \exp(-N\bar{x}/\lambda)\, f(\lambda).$$

We can make this much clearer by considering a numerical example. Suppose we are sampling from an exponential distribution with mean $\lambda = 7$.

Let's begin by generating some samples :

In[ ]:=
```
SeedRandom[1234]; λtrue = 17.0; nsample = 30;
xAll = RandomVariate[ExponentialDistribution[1 / λtrue], {nsample}]
```

Out[ ]=
```
{2.23881, 11.0527, 41.6638, 16.5426, 75.6996, 1.28375, 10.3573,
 12.5012, 23.8862, 4.66775, 0.257053, 25.9701, 13.2373, 2.08205,
 9.14777, 22.6424, 1.42562, 14.593, 0.21745, 9.02914, 42.3516, 4.00266,
 6.15701, 4.84835, 15.5112, 7.78079, 3.2828, 38.2276, 12.7898, 19.9101}
```

Now we define a selection range of the instrument and choose only data that lie within that range. For illustrative purposes we choose slightly different values from the David MacKay example:

In[ ]:=
```
xmin = 1; xmax = 30;
x = Select[xAll, xmin < # < xmax &]
```

Out[ ]=
```
{2.23881, 11.0527, 16.5426, 1.28375, 10.3573, 12.5012, 23.8862, 4.66775,
 25.9701, 13.2373, 2.08205, 9.14777, 22.6424, 1.42562, 14.593, 9.02914,
 4.00266, 6.15701, 4.84835, 15.5112, 7.78079, 3.2828, 12.7898, 19.9101}
```

The likelihood of the first n samples of the data is:

In[ ]:=
```
likelihood[λ_, n_, xmin_, xmax_] :=
 1 / ((Exp[-xmin / λ] - Exp[-xmax / λ]) λ)^n Exp[-Sum[x[[i]] / λ, {i, 1, n}]]
```

A characteristic of the likelihood functions that arise in Bayesian probability theory is that they often lead to numerical underflow. In order to avoid this difficulty we work on a log scale instead and employ the log-max trick in the numerical computations that follow. This means we compute the log likelihood, subtract the max log likelihood and then exponentiate to obtain a normalized likelihood without underflow problems. If we need to make an evidence calculation (which we do not here) the maximum likelihood value can be added to the log evidence at the end of the computation.

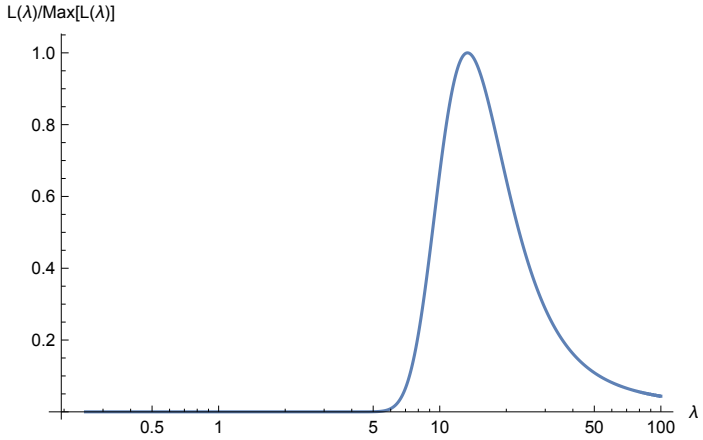
The log of the likelihood function is:

In[ ]:=
```
logLikelihood[λ_, n_, xmin_, xmax_] :=
 -n * Log[(Exp[-xmin / λ] - Exp[-xmax / λ]) λ] - Sum[x[[i]] / λ, {i, 1, n}]
```

The normalized likelihood of the data looks like the following plot. In doing this we also define a range for the unknown parameter λ as well as a sampling increment:

In[ ]:=
```
λmin = 0.25; λmax = 100.0; dλ = 0.025;
loglikely = Table[logLikelihood[λ, Length[x], xmin, xmax], {λ, λmin, λmax, dλ}];
loglikely = loglikely - Max[loglikely];
likely = Exp[loglikely];
ListLogLinearPlot[likely, ...]
```

Out[ ]=

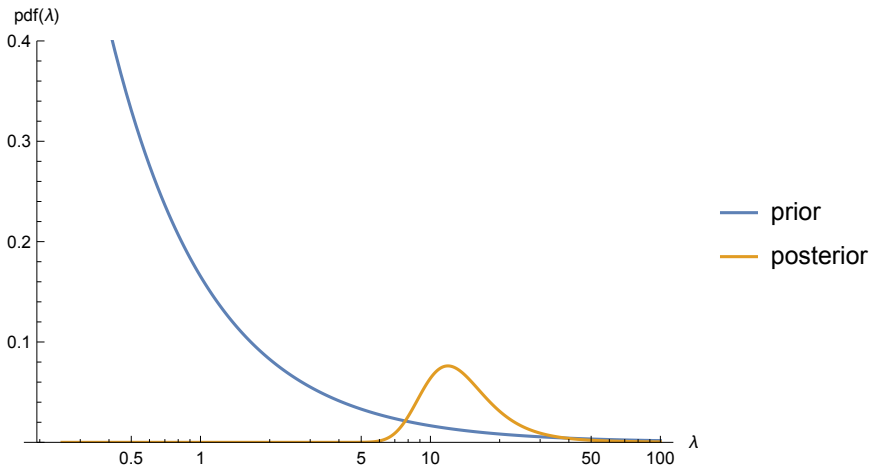


We assume a Jeffrey prior on λ on the range $\lambda_{min} < \lambda < \lambda_{max}$ of the form $f(\lambda) \propto \lambda^{-1}$. We do not need to express the constant of proportionality since it normalizes out in the numerical computation.

The prior and the posterior are plotted in the following:

In[ ]:=
```
dλ = 0.025;
prior = Table[λ^-1, {λ, λmin, λmax, dλ}];
prior = prior / Total[prior];
posterior = prior * likely;
posterior = posterior / Total[posterior];
ListLogLinearPlot[{prior, posterior} / dλ, ···]
```

Out[ ]=



The posterior mean is:

In[ ]:=
```
posterior.Table[λ, {λ, λmin, λmax, dλ}]
```

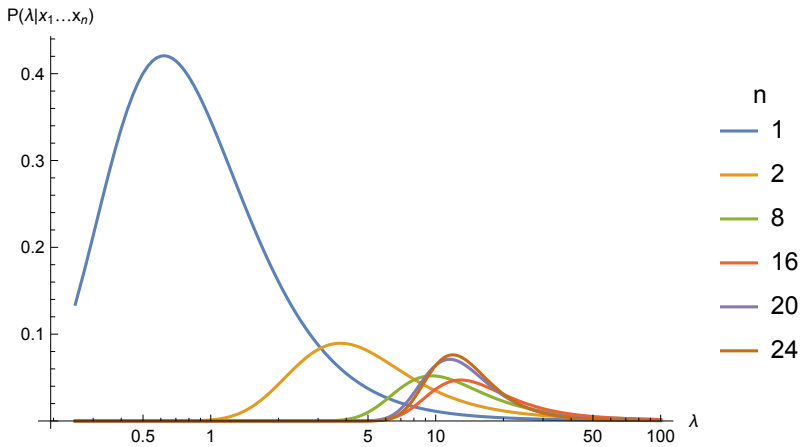
Out[ ]=

```
19.8766
```

As sample size increases the posterior probability density function becomes more tightly centered on $\lambda_{true} = 17$:

In[ ]:=
```
nchoice = {1, 2, 8, 16, 20, 24};
posteriors = Table[
    likelyn = Table[logLikelihood[λ, n, xmin, xmax], {λ, λmin, λmax, dλ}];
    likelyn = Exp[likelyn - Max[likelyn]];
    posterior = prior * likelyn;
    posterior = posterior / Total[posterior];
    posterior, {n, nchoice}];
ListLogLinearPlot[posteriors / dλ, ···]
```

Out[ ]=



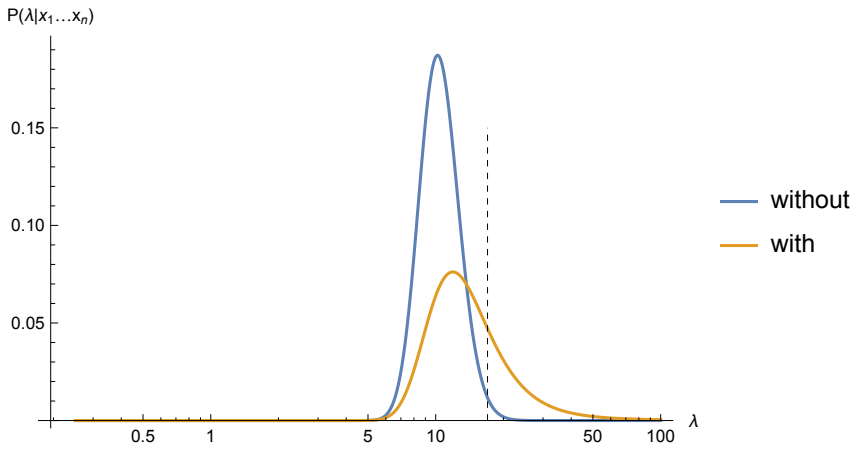
If we did not include selection effects then the log likelihood of the data would be:

In[ ]:=
```
logLikelihoodNS[λ_, n_, xmin_, xmax_] := -n * Log[λ] - Sum[x[[i]] / λ, {i, 1, n}];
```

The following plot compares the posteriors computed with and without selection effects treated. The location of λ_{true} is indicated by the vertical dashed line:

In[ ]:=
```
loglikelyNS = Table[logLikelihoodNS[λ, Length[x], xmin, xmax], {λ, λmin, λmax, dλ}];
loglikelyNS = loglikelyNS - Max[loglikelyNS];
likelyNS = Exp[loglikelyNS];
posteriorNS = likelyNS * prior;
posteriorNS = posteriorNS / Total[posteriorNS];
ListLogLinearPlot[{posteriorNS, Last[posteriors]} / dλ, ···]
```

Out[ ]=



In the foregoing analysis we were able to treat selection effects because we were able to write down an expression for the probability density function that was generating the observed data. This is not always possible or convenient. In the material that follows we present an alternate approach that does not require this.

Universal PDF of detected SNR

In the following discussion we derive the universal probability density function for the distribution of detected *SNR* values and show how this probability density function (PDF) can be used to estimate the slant range decay rate of radar echoes due to reflection from clear air scatter. By decay rate we mean the way in which radar echoes diminish in power as a function of slant range *r.* Specifically we consider the case in which the decay as a function of slant range is given by r^{-a} where a is a constant. The approach that we take here is adapted from Wen and Schutz (2012) who were concerned with the passive detection of gravitational waves. From the stand point of signal processing, there are strong similarities between the passive detection of gravitational waves and the active detection of radar echoes. By detected signal to noise ratio (*SNR*) we refer to radar echoes that are above a threshold that is large enough to insure that the echo is produced by reflection from a target and is not just a spike in the background noise level.

The *SNR* of a radar echo can be written in the simplified form

$$y = F / r^a$$

where the symbol y denotes the *SNR* of the echo measured on a power scale, F is the figure of merit of the radar, r is the radial distance (or slant range) to the target and a is the decay rate constant. The

cases $a = 1$, 2 and 4 respectively refer to the passive detection of an advancing cylindrical wave, the weather radar equation and the ordinary radar equation. For this last case Blake (1991)

$$F = t_p \frac{P_t G_t G_r \sigma_{\text{RCS}} \lambda^2}{(4\pi)^3 k_B T_{\text{sys}}}$$

where t_p is the radar pulse length, P_t is the transmit power, G_t is the gain of the transmit antenna, G_r is the gain of the receive antenna, σ_{RCS} is the radar cross section of the target, λ is the wavelength of the radar carrier frequency, k_B is Boltzmann's constant and T_{sys} is the noise temperature of the radar. If y_T denotes the threshold SNR for the radar, then the maximum range at which the radar can make a detection is

$$r_{\max} = F^{1/a} \Big/ y_T^{1/a}$$

If β denotes the constant density per unit volume of targets, then the number of detections produced by the radar at the two SNR thresholds y_T and $y > y_T$ is

$$N_D(y_T) = \beta \Omega_{\text{beam}} F^{3/a} y_T^{-3/a}, \quad N_D(y) = \beta \Omega_{\text{beam}} F^{3/a} y^{-3/a}$$

where Ω_{beam} is the solid angle subtended by the radar beam. If the radar is omnidirectional and the entire hemisphere is illuminated then $\Omega_{\text{beam}} = 2\pi$. The fraction of targets that are detected at *SNR* values $y > y_T$ is the ratio of these two quantities

$$\text{fraction detected} = (y_T / y)^{3/a}$$

If $y = y_T$ then the fraction detected is unity. The cumulative fraction of targets detected as a function of the *SNR* value y is

$$G(y) = \begin{cases} 1 - (y_T / y)^{3/a} & y \geq y_T \\ 0 & y < y_T \end{cases}$$

At this point it is convenient to make a change in notation that simplifies the analysis that follows. The change that we make is to define

$$\alpha = 3/a$$

As the parameter a assumes the values 1, 2 and 4 then α becomes 3, 3/2 and 3/4. With this change in notation, the cumulative fraction of targets detected as a function of the *SNR* value y is

$$G(y) = \begin{cases} 1 - (y / y_T)^{-\alpha} & y \geq y_T \\ 0 & y < y_T \end{cases}$$

The function $G(y)$ is the cumulative probability density function of a Pareto random variable with probability density function $g(y)$

$$g(y) = \frac{d}{dy} G(y) = \begin{cases} \alpha y_T^{\alpha} y^{-\alpha-1} & y \geq y_T \\ 0 & y < y_T \end{cases}$$

When $\alpha = 3$, corresponding to the scalar (not power) passive detection of an advancing gravity wave or the passive power detection of a cylindrically spreading acoustic wave, then $g(y) = 3 y_T^3 y^{-4}$ for $y \geq y_T$ and zero otherwise. This is precisely the form obtained by obtained by Wen and Schutz (2012). They refer to $g(y)$ as the *universal probability density function for the distribution of detected SNR values*. The probabil-

ity density function $g(y)$ does not depend upon the figure of merit F of the radar. The range dependence in the underlying in the underlying radar equation that governs the propagation physics is encoded in the parameter $\alpha = 3/a$. The median value of a Pareto distribution with threshold y_T and decay constant α is $2^{1/\alpha}\, y_T$. This implies that 1/2 of all detections will occur in the SNR range $y_T \le y \le 2^{1/\alpha}\, y_T$. For the case $\alpha = 3/2$ (weather radar equation with $a = 2$), 1/2 of detections occur within $10 \log_{10} 2^{2/3} = 2$ dB of the threshold y_T.

Numerical example

In order to perform some interesting simulations we will assume that target range and target reflectivity are random quantities. Given that β is the number of targets per unit volume, then the number of targets in the distance band $(r, r + dr)$ is $\beta(4/3)\,\pi\left((r + dr)^3 - r^3\right) \approx 4\,\pi r^2\,\beta dr$. This implies that the probability density and cumulative density of targets as a function of range are

$$h(r) = \frac{3\,r^2}{r_{\max}^3 - r_{\min}^3}, \quad H(r) = \frac{r^3 - r_{\min}^3}{r_{\max}^3 - r_{\min}^3}$$

where $r_{\min} < r < r_{\max}$ is limiting range of targets. The range of targets can be simulated by computing

$$r = \left(r_{\min}^3 + \left(r_{\max}^3 - r_{\min}^3\right) u\right)^{1/3}$$

where u is uniformly distributed on the interval (0, 1). Regarding the distributions of target reflectivity, we will assume that these quantities are either uniformly distributed or log uniformly distributed on the range $(\sigma_{\min}, \sigma_{\max})$. This last choice is known as a Jeffreys prior and the probability density function is

$$q(\sigma) = \frac{1}{\log(\sigma_{\max}/\sigma_{\min})}\ \sigma^{-1}, \quad \sigma_{\min} < \sigma < \sigma_{\max}$$

and zero otherwise.

The following module generates synthetic SNR data above the threshold y_T. Targets can be uniformly distributed in range or in volume. The target radar cross section can be uniformly distributed on the interval ($\sigma_{\min}$(dB) $\sigma_{\max}$(dB)) or log log uniform on the interval:

In[ ]:=
```
MakeDataGWPPR[yT_, targetRCSPDF_, nMonteCarlo_,
    σMinDB_, σMaxDB_, rMin_, rMax_, targetAltitudePDF_] :=
  Module[{Pt = 3.5, Gt = 10^3.7, Gr = 10^3.7, λ = 0.009, kB = 1.38 * 10^-23, Tsys = 300.0,
    tp = 190.0 * 10^-6, Cr4thPower, σMin, σMax, dataAllSNR, σRCS, r, dataDetectedSNR},
   Cr4thPower = tp Pt * Gt * Gr * λ^2 / ((4 π)^3 kB * Tsys);
   σMin = 10^(σMinDB/10.0); σMax = 10^(σMaxDB/10.0);
   dataAllSNR = Table[
     If[targetRCSPDF == "UniformPDF", σRCS = RandomReal[{σMin, σMax}]];
     If[targetRCSPDF == "JeffreysPFD", σRCS = RandomReal[{σMinDB, σMaxDB}];
      σRCS = 10^(σRCS/10)];
     If[targetAltitudePDF == "UniformInAltitude", r = RandomReal[{rMin, rMax}]];
     If[targetAltitudePDF == "UniformInVolume",
      r = (rMin^3 + (rMax^3 - rMin^3) RandomReal[])^1/3];
     {r, Cr4thPower σRCS / r^4}, {nMonteCarlo}];
   dataDetectedSNR = Select[dataAllSNR, #[[2]] > yT &];
   dataDetectedSNR];
```

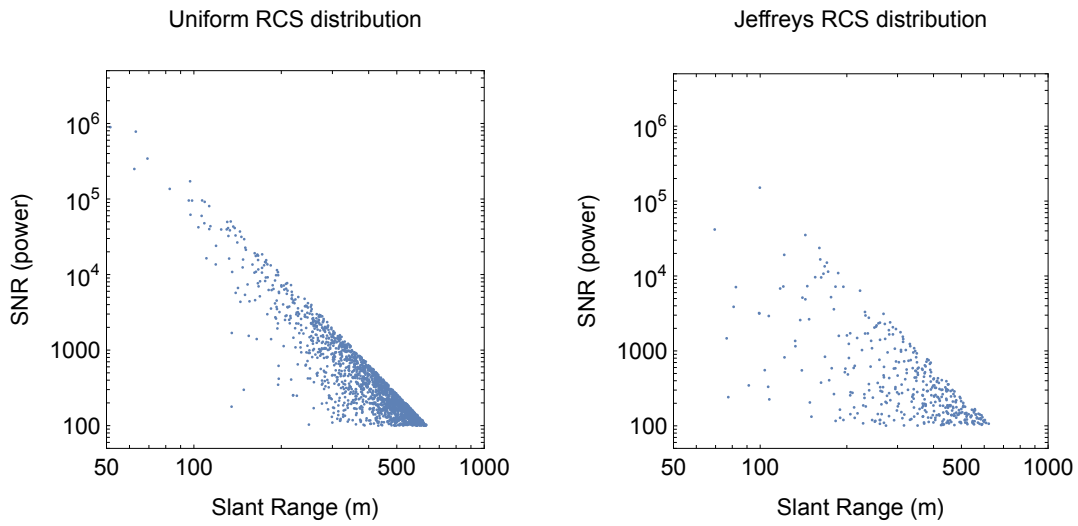
In[ ]:=

Assume targets are uniformly distributed in volume

We begin by generating synthetic data assuming either a uniform distribution for target reflectivity or log uniform. Target range is assumed to be uniformly distributed in volume for either case. First we look at how detected SNR varies in range:

In[ ]:=
```
SeedRandom[12 349 876];
nMonteCarlo = 100 000;
targetAltitudePDF = "UniformInVolume";
yT = 100.0; σMinDB = -90.0; σMaxDB = -40.0; rMin = 50; rMax = 2000;
dataUniform = MakeDataGWPPR[yT, "UniformPDF",
   nMonteCarlo, σMinDB, σMaxDB, rMin, rMax, targetAltitudePDF];
dataJeffreys = MakeDataGWPPR[yT, "JeffreysPFD",
   nMonteCarlo, σMinDB, σMaxDB, rMin, rMax, targetAltitudePDF];
gUniform = ListLogLogPlot[dataUniform, ···];
gJeffreys = ListLogLogPlot[dataJeffreys, ···];
GraphicsRow[{gUniform, gJeffreys}]
```

Out[ ]=



Since we assumed that echoes fall like r^{-4} then we would expect that our simulated data would fit a Pareto distribution with parameter $\alpha = 3/4$. For the assumption of the uniform distribution of target radar cross section we find:

In[ ]:=
```
dataUniformRCS = Map[Last, dataUniform];
estU = EstimatedDistribution[dataUniformRCS, ParetoDistribution[k, α]]
```

Out[ ]=
```
ParetoDistribution[100.003, 0.747054]
```

The agreement is quite good with $y_t = 100$ and $\alpha = 3/4$.

The estimated decay with range is:

In[ ]:=
```
3 / estU[[2]]
```

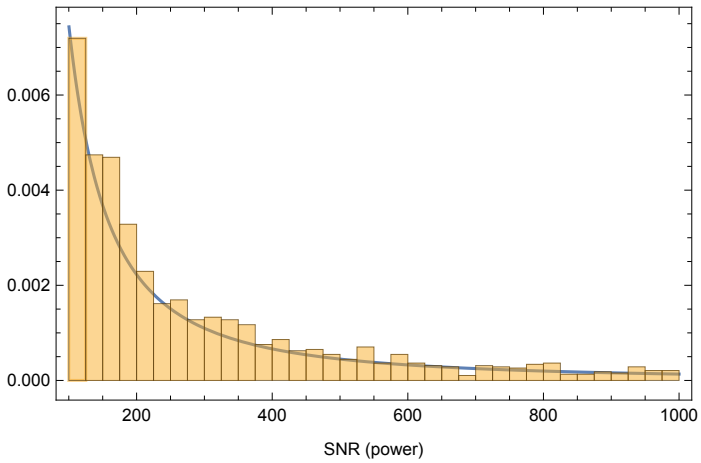
Out[ ]=
```
4.01577
```

which is almost exactly 4.

The histogram of the synthetic data compared to the fitted probability density function is:

In[ ]:=
```
g1 = Histogram[{dataUniformRCS}, ...];
g2 = Plot[PDF[estU, y], {y, 100, 1000}, ...];
Show[g2, g1]
```

Out[ ]=



For the assumption of the Jeffreys distribution of target radar cross section we find:

In[ ]:=
```
dataJeffreysRCS = Map[Last, dataJeffreys];
estJ = EstimatedDistribution[dataJeffreysRCS, ParetoDistribution[k, α]]
```

Out[ ]=
```
ParetoDistribution[100.665, 0.77536]
```

Again the agreement between the simulation parameters and the recovered parameters is good.

The estimated decay with range is:

In[ ]:=
```
3 / estJ[[2]]
```

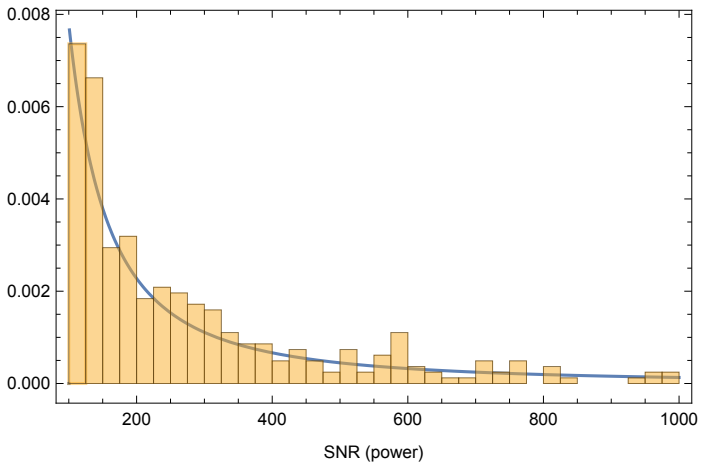
Out[ ]=
```
3.86917
```

which is very nearly 4.

The histogram and fitted probability density function are:

In[ ]:=
```
g1 = Histogram[{dataJeffreysRCS}, ...];
g2 = Plot[PDF[estJ, y], {y, 100, 1000}, ...];
Show[g2, g1]
```

Out[ ]=



In[ ]:=

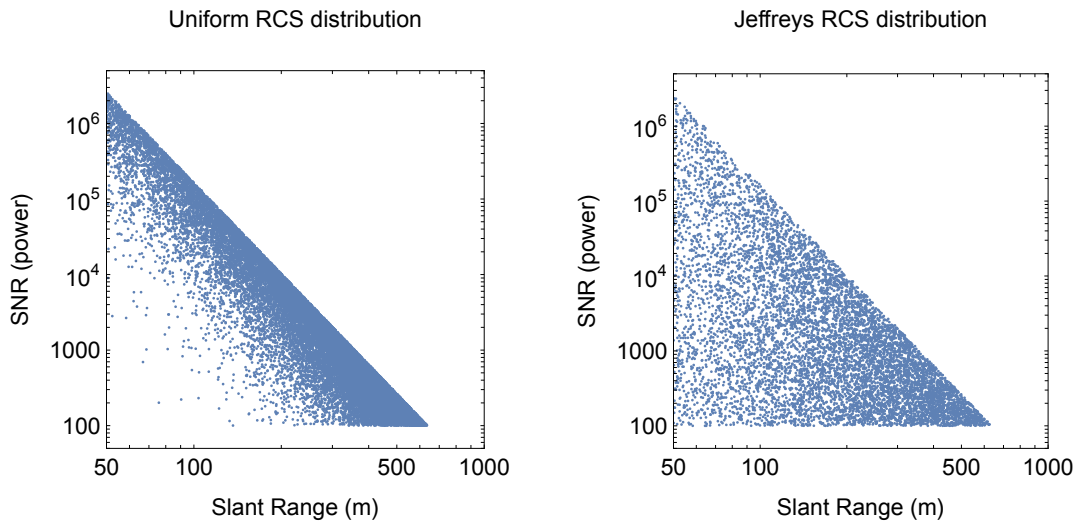
Assume targets are uniformly distributed in altitude

Now we assume that the targets are uniformly distributed in altitude. This breaks one of our fundamental assumptions in the initial model and we cannot analyze the distribution of detected SNR to determine the rate of range decay.

We begin by computing:

In[ ]:=
```
SeedRandom[12 349 876];
nMonteCarlo = 100 000;
targetAltitudePDF = "UniformInAltitude";
yT = 100.0; σMinDB = -90.0; σMaxDB = -40.0; rMin = 50; rMax = 2000;
dataUniform = MakeDataGWPPR[yT, "UniformPDF",
   nMonteCarlo, σMinDB, σMaxDB, rMin, rMax, targetAltitudePDF];
dataJeffreys = MakeDataGWPPR[yT, "JeffreysPFD",
   nMonteCarlo, σMinDB, σMaxDB, rMin, rMax, targetAltitudePDF];
gUniform = ListLogLogPlot[dataUniform, ···];
gJeffreys = ListLogLogPlot[dataJeffreys, ···];
GraphicsRow[{gUniform, gJeffreys}]
```

Out[ ]=



Since we assumed that echoes fall like r^{-4} then we would hope that our simulated data would fit a Pareto distribution with parameter $\alpha = 3/4$. For the assumption of the uniform distribution of target radar cross section we find:

In[ ]:=
```
dataUniformRCS = Map[Last, dataUniform];
estU = EstimatedDistribution[dataUniformRCS, ParetoDistribution[k, α]]
```

Out[ ]=
```
ParetoDistribution[100.084, 0.33312]
```

The agreement is quite not good with $y_t = 100$ and $\alpha = 3/4$.

For the assumption of the Jeffreys distribution of target radar cross section we find:

In[ ]:=
```
dataJeffreysRCS = Map[Last, dataJeffreys];
estJ = EstimatedDistribution[dataJeffreysRCS, ParetoDistribution[k, α]]
```

Out[ ]=
```
ParetoDistribution[100.015, 0.382633]
```

Again the agreement between the simulation parameters and the recovered parameters is not good .

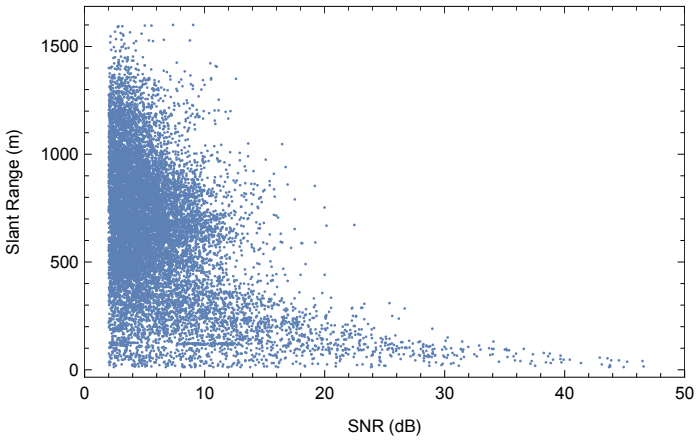
Observation: If targets are uniformly distributed in volume then the distribution of detected SNR can be analyzed (fitted with a Pareto distribution) to infer the rate of range decay. However if the distribution of targets is uniform in altitude then the technique fails.

Application to real data

The following figure shows some WiPPR data recorded at Yuma AZ in 2017. The SNR threshold is 2 dB:

In[ ]:=
```
rangeDecay = TimeSeries[ <<Time: 2. to 46.6, Data points: 12 806>> ];
ListPlot[rangeDecay, <<...>>]
```

Out[ ]=



Lets fit a Pareto distribution:

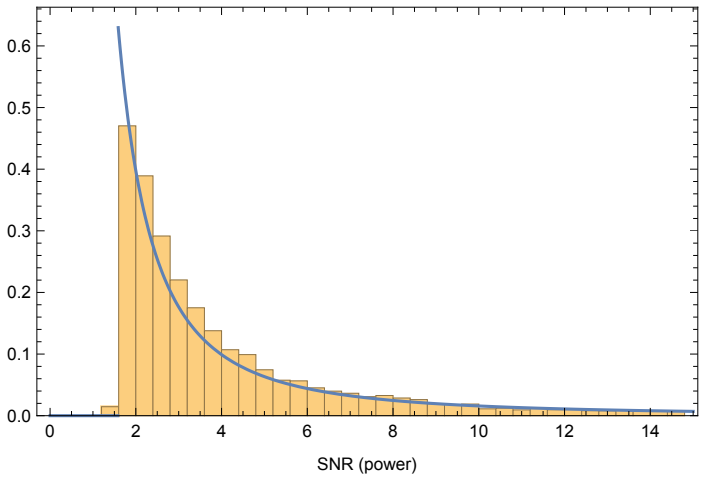
In[ ]:=
```
data = rangeDecay["Times"];
data = 10^(data/10);
estdist = EstimatedDistribution[data, ParetoDistribution[k, α]]
```

Out[ ]=
```
ParetoDistribution[1.58491, 1.00399]
```

The histogram and fitted probability density function are:

In[ ]:=
```
g1 = Histogram[data, {0, 15, 0.4}, ...];
g2 = Plot[PDF[estdist, y], {y, 0, 15}, ...];
Show[g1, g2]
```

Out[ ]=



The implied range decay coefficient is $a = 3/\alpha$ where α is the fitted Pareto parameter:

In[ ]:=
```
3 / estdist[[2]]
```

Out[ ]=
```
2.98807
```

The value is almost exactly 3.This implies $30 \log_{10} r$ SNR range decay on a decibel scale in the data. This is not unreasonable. This suggests that the range decay in the data is mid way between the weather radar equation ($20 \log_{10} r$ range decay) and the standard radar equation ($40 \log_{10} r$ range decay). An advantage of the technique is that it makes use of all the data and properly accounts for selection effects (thresholding).

References

MacKay, David (2003), *Information Theory, Inference and Learning Algorithms*, Cambridge University Press.

Wen, L. and Schutz, B. F. (2012), "Chapter 5: Network analysis and multi-messenger astronomy", published in *Advanced Gravitational Wave Detectors*, edited by D. G. Blair. E. J. Howell, L. Ju and C. Zhao, Cambridge University Press, first published 2012.

Author notes

The figure in the first section was produced with the following code:

```
SeedRandom[1236];
instrument = {Gray, Rectangle[{-1.2, -2}, {-0.2, 2}], Rectangle[{-1, -1}, {-2, 1}]};
```

```
data = {PointSize[0.015], Map[Point, Sort[Table[{RandomReal[{1, 20}], 0}, {7}]]]};
ref = {Arrow[{{0.2, -1}, {7, -1}}]}; lbl = Text["x", {8, -1}];
lbl1 = Text["x=1", {2.1, 2}]; lbl2 = Text["x=20", {21.1, 2}];
barrier1 = {Dashing[0.01], Line[{{1, -2}, {1, 2}}]}; barrier2 = {Dashing[0.01],
Line[{{20, -2}, {20, 2}}]};
Graphics[{instrument, data, lbl, barrier1, barrier2, ref, lbl1, lbl2}, PlotRange ->
{{-3, 23}, All}]
```