A Bayesian approach to distribution fitting

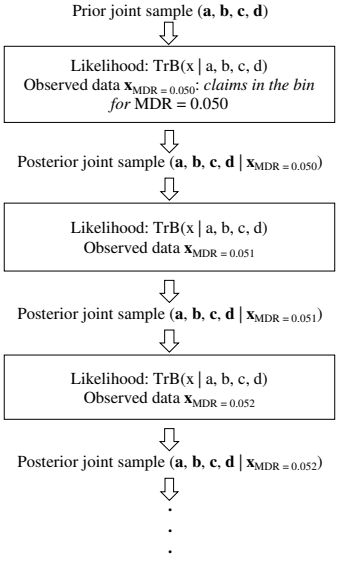
Objective: Frame the problem in Bayesian context.

Foundation: Locality of distributions along the MDR axis.

1. Close MDRs $\Longrightarrow$ similar TrB PDFs $\Longrightarrow$ similar TrB parameters.
2. More than just assumptions: smooth transition is required in distributions along the MDR axis.
3. Posterior from the nearest MDR is used as prior.

Algorithms:

1. Kalman filter.
   - **Unknown** transformation of (a, b, c, d) but **known** likelihood — TrB.
   - Not obviously suitable: non-Gaussian likelihood + posterior.
   - Impose Gaussianity. Use posterior sample mean + covariance for reparameterization.
     - Borrow from the idea of Unscented Transform.



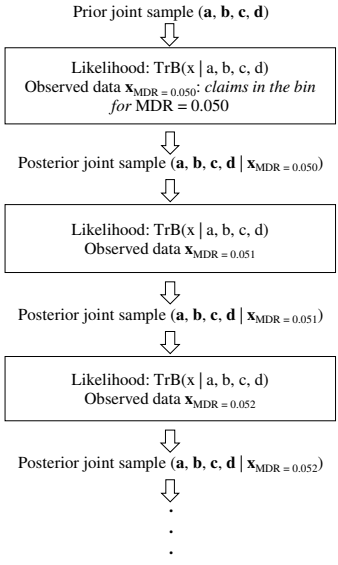
A Bayesian approach to distribution fitting

Algorithms:

2. Particle filter.
   - Sample Impoverishment. Remedies:
     - Resample less frequently.
     - Sample roughening: add random noise to resampled particles [REF].
     - MCMC (Markov Chain Monte Carlo). Extremely slow due to long chain, unless we infer parametric posterior for every new MDR.
   - More on combating Sample Impoverishment:
     - For 1-d case, one could first compute a histogram from the weighted particles, and then sample from it assuming uniform distribution within bins.
     - For high-d posterior, use Metropolis Hastings / Gibbs to draw sample one at a time. Remove burn-in afterwards.
     - Hamilton MCMC, NUTS (No U-Turn Sampler) require fewer samples to faithfully represent the full distribution but demand gradients of the posterior during computation. Not popular until auto-differentiation libraries for deep learning matured in mid 2010s.

Point estimate:

From posterior samples of (a, b, c, d), select the one that maximizes the posterior likelihood.



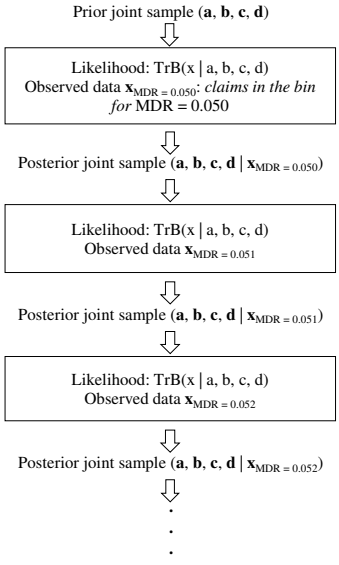
A Bayesian approach to distribution fitting

General steps:

1. Fit and smooth P_0.
2. Fit TrB distributions to the nonzero claim damage ratios. Constraint:

$$\mathbb{E}(\min(X, 1)) = \frac{\text{MDR}}{1 - P_0}$$

3. Discretization. PDF $\Longrightarrow$ PMF.
4. Fine-tuning.
   - Match PMF's mean to MDR, ensure monotonicities, etc.



Incorporate constraint of limited mean, approach I

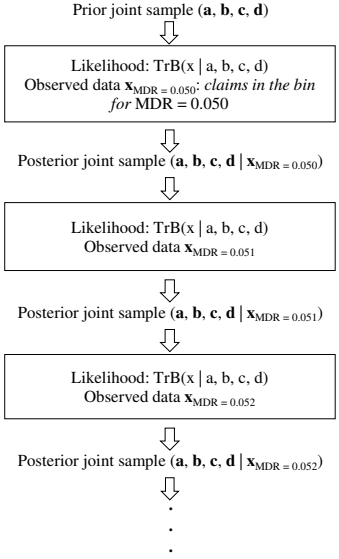
Consider $X|(a,b,c,d) \sim$TrB:

$$\begin{aligned}\mu_{\text{lim}} &= \mathbb{E}(\min(X,1)|a,b,c,d) \\ &= \frac{d\cdot\Gamma\left(\frac{b+1}{c}\right)\Gamma\left(\frac{a-1}{c}\right)}{\Gamma\left(\frac{a}{c}\right)\Gamma\left(\frac{b}{c}\right)}\cdot\beta\left(\frac{b+1}{c},\frac{a-1}{c};\frac{1}{d^c+1}\right) \\ &+1-\beta\left(\frac{b}{c},\frac{a}{c};\frac{1}{d^c+1}\right) \text{ where}\end{aligned}$$

$$\beta(u,v;x)=\frac{\Gamma(u+v)}{\Gamma(u)\Gamma(v)}\int_0^x t^{u-1}(1-t)^{v-1}dt\ .$$

[Klugman 2019, A.2.1.1].

Given a prior joint PDF for (a,b,c,d), deriving the PDF of μ_{lim} directly is *hard*.



Incorporate constraint of limited mean, approach I

Bayesian nonlinear regression:

$$\text{Let } \mu_{\text{lim}} \sim \mathcal{N}(m, \sigma^2) \text{ where}$$

$$m = \frac{d \cdot \Gamma\left(\frac{b+1}{c}\right) \Gamma\left(\frac{a-1}{c}\right)}{\Gamma\left(\frac{a}{c}\right) \Gamma\left(\frac{b}{c}\right)} \cdot \beta\left(\frac{b+1}{c}, \frac{a-1}{c}; \frac{1}{d^c+1}\right) + 1 - \beta\left(\frac{b}{c}, \frac{a}{c}; \frac{1}{d^c+1}\right) ,$$

$$\beta(u, v; x) = \frac{\Gamma(u+v)}{\Gamma(u)\Gamma(v)} \int_0^x t^{u-1}(1-t)^{v-1} dt ,$$

σ^2 is predefined.

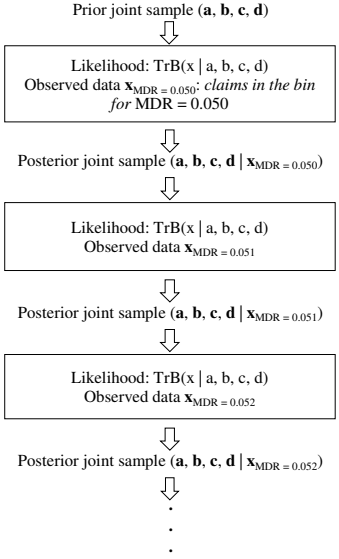
Assumption: μ_{lim} and X (claim damage ratio) are conditionally independent on (a, b, c, d) , then

$$\Pr(a, b, c, d | x, \mu_{\text{lim}}) \propto \Pr(a, b, c, d) \cdot \Pr(x | a, b, c, d) \cdot \Pr(\mu_{\text{lim}} | a, b, c, d)$$

In a MDR interval, the observed μ_{lim} are constant $= \text{MDR}/(1 - P_0)$.

The approach is more or less equivalent to Bayesian optimization: for samples of (a, b, c, d) that produce μ_{lim} closer to $\text{MDR}/(1 - P_0)$, we reward them with higher weight (probability). The reward function is $\propto$ a Gaussian kernel of $\|$sampled μ_{lim} - $\text{MDR}/(1 - P_0)\|$.

Comparing to the frequentist approach, this corresponds to an objective function that combine error in mean and distance between PMFs.



Incorporate constraint of limited mean, approach II

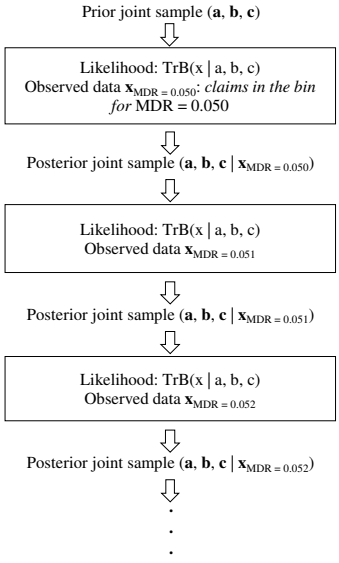
Given observed μ_{lim} for a sampled (a, b, c), solve d from:

$$\mu_{\text{lim}} = \frac{d \cdot \Gamma\left(\frac{b+1}{c}\right)\Gamma\left(\frac{a-1}{c}\right)}{\Gamma\left(\frac{a}{c}\right)\Gamma\left(\frac{b}{c}\right)} \cdot \beta\left(\frac{b+1}{c}, \frac{a-1}{c}; \frac{1}{d^c+1}\right) + 1 - \beta\left(\frac{b}{c}, \frac{a}{c}; \frac{1}{d^c+1}\right)$$

μ_{lim} is an increasing (?) function of d . Use bisection to solve for d on the fly.

Newton-Raphson might be worth a try since $\partial\mu_{\text{lim}}/\partial d$ is not too difficult to compute. Requires extra care to bound the solution in $(0, +\infty)$.

Approach II removes the stochasticity in matching MDR. Preferred.



Summary

Bayesian advantages:

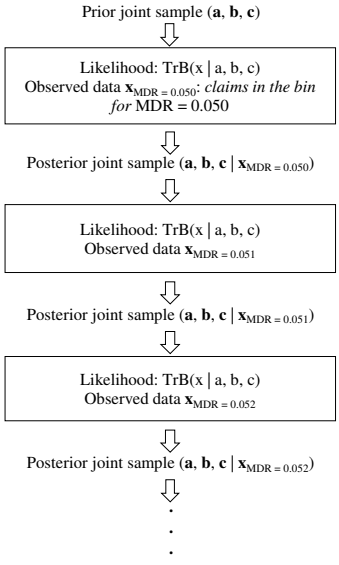
- Robust.
- Offer richer statistics, e.g. parameter covariances, credible intervals.
- Ensure smooth transition in fitted distributions along the MDR axis.

Disadvantages:

- Slow.
- Choosing prior can be subjective. Non-informative priors will unlikely lead to different results from the frequentist approach.

Notes:

1. We still take only a point estimate, e.g., posterior maximum likelihood estimator, as the final result.
   - If the frequentist optimizes (a, b, c, d) using the maximum likelihood objective, then the difference between Bayesian and the frequentist lies only in the optimization methods: deterministic vs. stochastic.
2. If catastrophe model predictions are poor, Bayesian will bring no magic.



Update distributions with historical TrB parameters and new claims data

Use Japan earthquake as an example. Let $\boldsymbol{x}_{\text{Japan2011}}$ be the Tohoku claims data. Let $\boldsymbol{x}_{\text{Japan2023}}$ be the claims data in a hypothetical event in Japan 2023.

Plain model:

$$p(a,b,c,d|\boldsymbol{x}_{\text{Japan2011}},\boldsymbol{x}_{\text{Japan2023}}) \propto p(a,b,c,d)\cdot p(\boldsymbol{x}_{\text{Japan2011}},\boldsymbol{x}_{\text{Japan2023}}|a,b,c,d)$$

This model ignores historical TrB parameters. The new estimates could fall far away from the historical, posing challenges to change management.

Preferred model: $p(a,b,c,d|a_0,b_0,c_0,d_0,\boldsymbol{x}_{\text{Japan2023}})$ where a_0,b_0,c_0,d_0 are historical point estimates of the TrB parameters.

$$\begin{aligned} p(a,b,c,d|a_0,b_0,c_0,d_0,\boldsymbol{x}_{\text{Japan2023}}) &\propto p(a,b,c,d)\cdot p(a_0,b_0,c_0,d_0,\boldsymbol{x}_{\text{Japan2023}}|a,b,c,d) \\ &= p(a,b,c,d)\cdot p(a_0,b_0,c_0,d_0|a,b,c,d)\cdot p(\boldsymbol{x}_{\text{Japan2023}}|a,b,c,d) \end{aligned}$$

Notice a_0,b_0,c_0,d_0 are not claims like $\boldsymbol{x}_{\text{Japan2023}}$, but are parameters that characterize a distribution of claims.

To evaluate $p(a_0,b_0,c_0,d_0|a,b,c,d)$, draw $\boldsymbol{x}'_{\text{Japan2011}} \sim \text{TrB}(a_0,b_0,c_0,d_0)$, then

$$p(a,b,c,d|a_0,b_0,c_0,d_0,\boldsymbol{x}_{\text{Japan2023}}) \propto p(a,b,c,d)\cdot p(\boldsymbol{x}'_{\text{Japan2011}},\boldsymbol{x}_{\text{Japan2023}}|a,b,c,d)$$

Update distributions with historical TrB parameters and new claims data

Model:

$$p(a,b,c,d|a_0,b_0,c_0,d_0,\boldsymbol{x}_{\text{Japan2023}}) \propto p(a,b,c,d)\cdot p(\boldsymbol{x}'_{\text{Japan2011}},\boldsymbol{x}_{\text{Japan2023}}|a,b,c,d)$$

$$\boldsymbol{x}'_{\text{Japan2011}} \sim \text{TrB}(a_0,b_0,c_0,d_0)$$

What should be the sample size for $\boldsymbol{x}'_{\text{Japan2011}}$?

A natural choice: $N\left(\boldsymbol{x}'_{\text{Japan2011}}\right) \leftarrow N\left(\boldsymbol{x}_{\text{Japan2011}}\right)$.

We adjust $N\left(\boldsymbol{x}'_{\text{Japan2011}}\right)$ to reflect our believes in previous estimates vs. current data.

For example, if we believe the 2011 data is twice as credible as the 2023 data, then let $N\left(\boldsymbol{x}'_{\text{Japan2011}}\right) \leftarrow 2N\left(\boldsymbol{x}_{\text{Japan2023}}\right)$

This gives full control of change management.

Prior editing (Van Leeuwen)

Prior (sample) editing mimics Metropolis Hastings (MH) with great risks.

MH *edit*s the previous sampled particle by adding random noise, then accepts or rejects the new particle based on its posterior probability.

- MH converges to the true posterior if the random noise has certain properties.
- MH is statistically correct because **the prior and likelihood PDFs are known**.

Van Leeuwen prior editing *edit*s previous particles and reevaluate them **using the likelihood PDF only, because the prior PDF is unknown — the prior distribution is characterized by particles**.

Over time, Van Leeuwen may push the posterior closer to the likelihood.

This could hide the incorrectness of the underlying physical model — when posterior equals likelihood, model output always matches observation.

Progress

1. Familiarized with probabilistic programming using `PyMC` in Python.
   - Great tool for Bayesian inferences.
   - Uses SOTA Hamilton / No-U-Turn MCMC for sampling.
2. Implemented a good number of R's fundamental functions using Python. The functions cover data wrangling, visualization, file manipulation. Made migration from R to Python easy. Good in the long run.
   - Python is on average more efficient than R, but some fundamental functions, e.g., `match`, in R can be much faster. R's core team really optimized those functions down to the bottom. For example, R's `match` employs the same hashing technique in bucket-sort, while Python just sorts and compares.
3. Familiarized with Python and C++ integration using package `pybind11`.

Moving average

First attempt:

- Took Debora/Emile's groupings of claims data and P_0.
- Run particle filtering (PF).

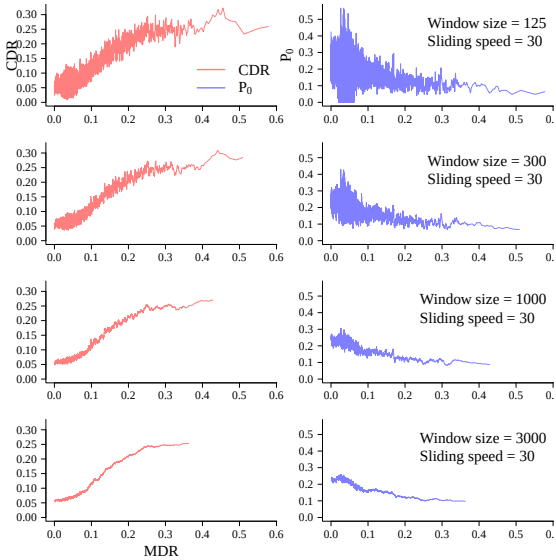
Non-overlapping MDR intervals are hard to make PF work.

- CDRs (claim damage ratios) vary too much from one interval to the next. Stable transition of parameters is hard to attain.
  - Debora/Emile's fitting also produces highly volatile parameters. Trends are then artificially imposed. Final distributions are largely detached from the originally fitted.
- Particles of (a, b, c) from the previous interval all have extremely small weights after likelihood evaluation in the current interval.
  - Analogous to the underlying physical model being completely off.

Sliding window moving average.

- Use a window for data collection and slowly pass it over along the MDR axis. High frequency signals attenuated; trends revealed.
- Every new window has only a small proportion of new data. Stable transition of parameters is more likely.

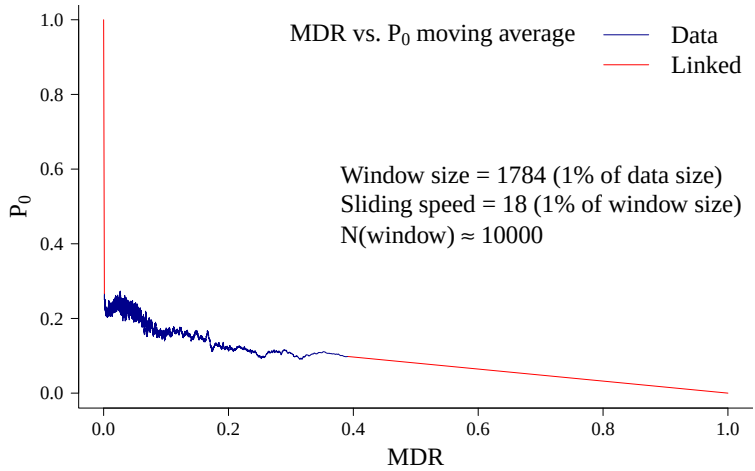
Optimal window size by analyzing MDR vs. CDR? Ongoing research.



Fit P_0

Constraint: P_0 is a decreasing function of MDR.

- $P_0(\text{MDR} = 0) = 1$
- $P_0(\text{MDR} = 1) = 0$



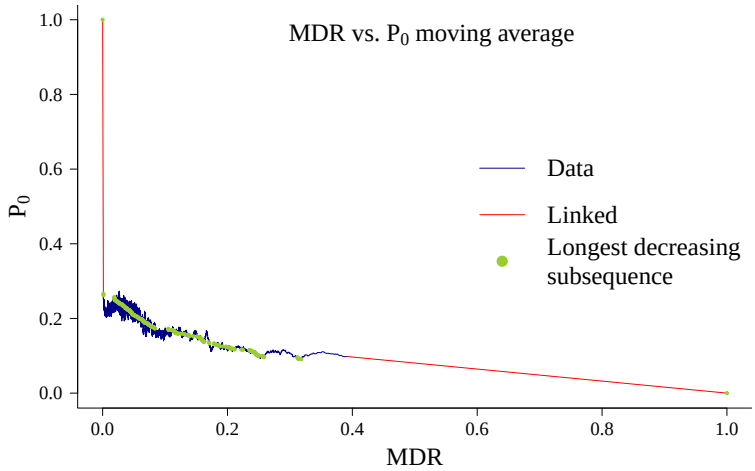
Fit P_0

Constraint: P_0 is a decreasing function of MDR.

- $P_0(\text{MDR} = 0) = 1$
- $P_0(\text{MDR} = 1) = 0$

Find the longest decreasing subsequence.

- A classic *dynamic programming* instance in algorithms.



Fit P_0

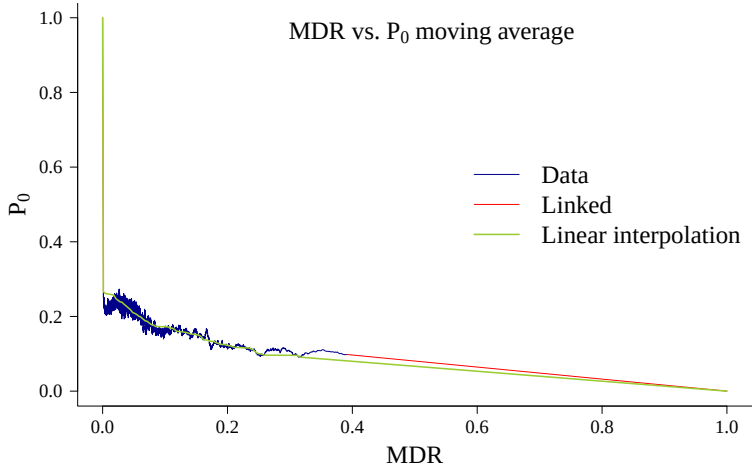
Constraint: P_0 is a decreasing function of MDR.

- $P_0(\text{MDR} = 0) = 1$
- $P_0(\text{MDR} = 1) = 0$

Find the longest decreasing subsequence.

- A classic *dynamic programming* instance in algorithms.

Approach I: Linear interpolation.



Fit P_0

Constraint: P_0 is a decreasing function of MDR.

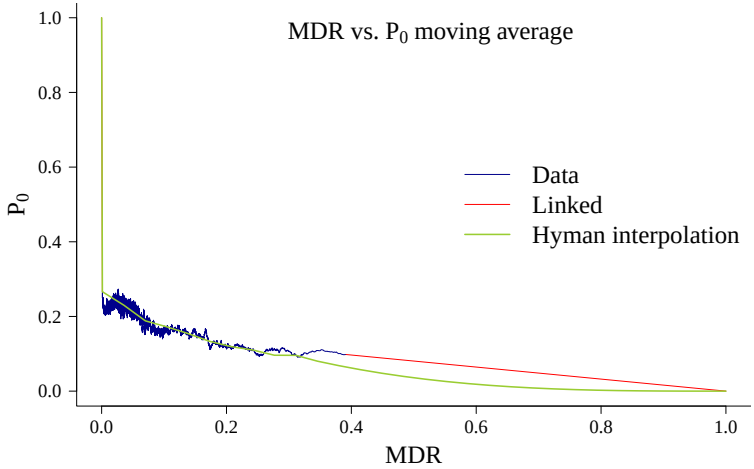
- $P_0(\text{MDR} = 0) = 1$
- $P_0(\text{MDR} = 1) = 0$

Find the longest decreasing subsequence.

- A classic *dynamic programming* instance in algorithms.

Approach I: Linear spline.

Approach II: Hyman convex cubic spline.



Fit P_0

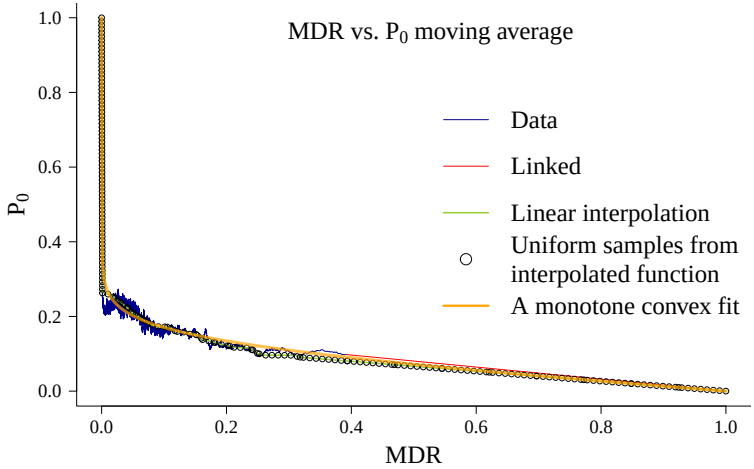
Approach III: Fully convex function in the form of

$$f(x) = \exp(-\beta)[1 - x^{\alpha}\exp(\beta x^{\gamma})] + 1 - \exp(-\beta)$$

Discovery:

1. The form is a modification on the CDF of the Tapered Pareto distribution. Researched during fire catalog downsampling.
2. Experiments show it is quite versatile for fitting convex, monotonically decreasing curves in various shapes.
3. Can be helpful with fitting parameters later if necessary.

For now, stick to Approach I (linear spine) as it is the most faithful to data.



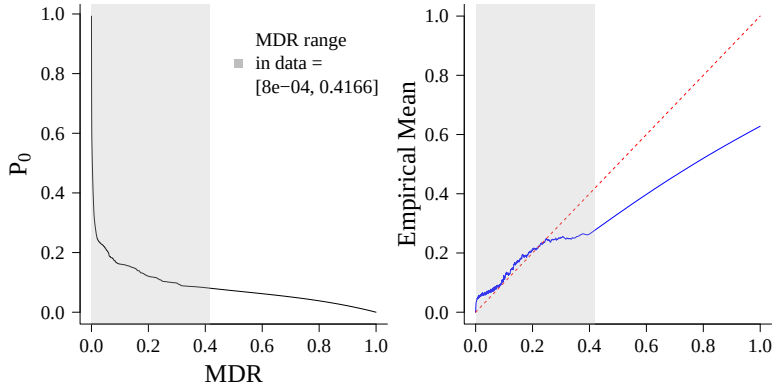
Architecture. Goal: minimize human intervention in training.

1. Import data.
2. Set sliding window size and speed.
   - Given by users. Recommendation in research. Autocorrelation? Elbow point?
3. Model P_0 via ensemble.
   - Sample K, e.g. 30 random subsets of data. Each subset is, e.g., 2/3 of the full data. For each subset:
   - Compute the sequence of P_0s in all windows.
   - Find the longest nondecreasing subsequence and fit it with smoothing spline.
   - Mean of the K splines is the final model.
4. Remove zero claims from data. Sample K random subsets again. For each subset:
5. Estimate 18999 empirical **cond**(**itional**) PMFs for all cond target MDRs.
   - PMFs for cond target MDR out of data range is extrapolated. We upscale (the support of) the PMF with the highest in-data MDR for the upper side, and downscale the PMF with the lowest in-data MDR for the bottom side.
   - The scaling is inherently adjusting parameter d **before** training. Previous methodology manipulates d for extrapolation **after** training.
   - Creating all 18999 empirical cond PMFs is necessary to make Bayesian update convenient and accurate.
5.
   - Users import 18999 old PMFs and define sampling weight w on them.
   - $P_0 \leftarrow (1-w)P_0 + wP_0^{\text{old}}$.
   - For each MDR, update the empirical cond PMF by mixing it with the old cond PMF.
6. Fit the 18999 empirical PMFs using quasi-Newton methods.
   - Dissected TrB and its derivatives with respect to parameters. Implemented multiple expansion series for computing thread-safe Regularized Incomplete Beta function. Implemented on-the-fly solver to parameter d inside objective function for imposing the limited mean (target MDR) – ∂limitedMean$/\partial d$ turns out simple enough and thus d is cheap to solve on the fly using Newton's.
   - Implemented Bayesian fitting via MCMC. Even with substantial effort for code optimization, it is still painfully slow comparing with quasi-Newton, making analysis too expensive. An evolutionary algorithm was also tried. Results from both are no better and often much worse than quasi-Newton, partly due to difficulty in finding optimal hyperparameters. But the idea of sequential update is still invaluable to smoothing TrB parameter transition.
   - Fully dissected the L-BFGS-B algorithm. Customized and optimized a competitive open-source C++ library for our problem setting. Implemented a number of distance measures between PMFs such as cross-entropy ($\equiv$ negative log-likelihood for weighted samples, aka PMF), Kolmogorov with smooth maximum (for differentiability), Euclidean CDF, etc. Tested multiple finite difference methods with numeric thresholds from publications for optimality.

Architecture. Goal: minimize human intervention in training.

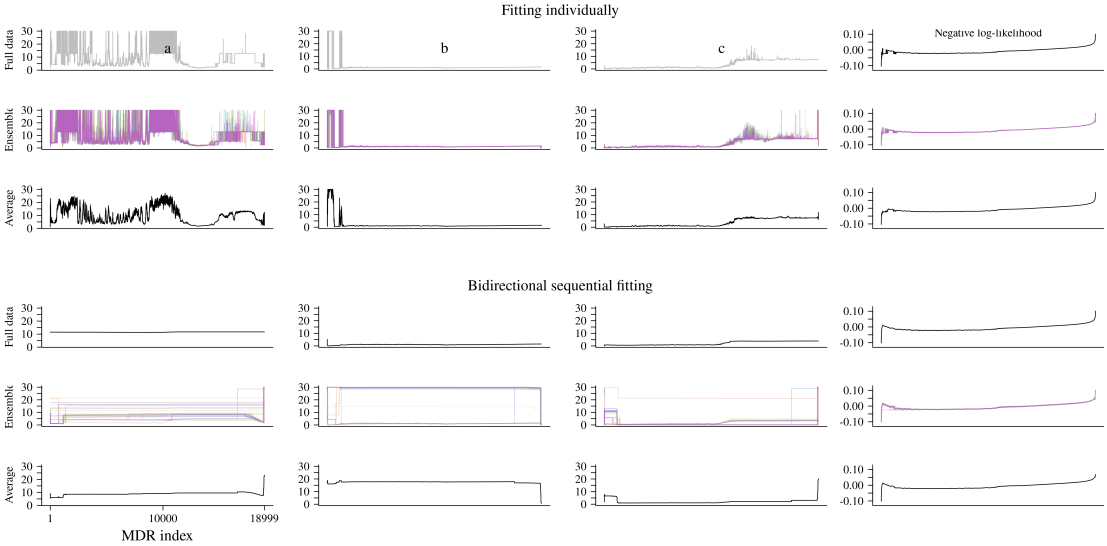
7. Findings and thoughts:
   - TrB is powerful. Experiments show it seems able to approach any right-skewed unimodal data distribution (even with rigid shape) with a well fitted parametric form.
   - TrB is highly numerically "pathological". Drastically different parameters might only make a small difference in the distribution's shape due to the nested exponentiations in the TrB parametric form. There are also symmetries to some extent among the parameters in certain regions. For example, in some cases two sets of quite different a and b in TrB can both well fit to the same data, because the tail (mainly controlled by a) and the general shape (mainly controlled by b) compensate each other.
   - Therefore, violent fluctuations in fitted TrB parameters along the MDR axis is mainly due to TrB's fitting power + numerical pathology, aka overfitting: the optimizer can push the parameters into a largely different area just to improve the goodness of fit by a little. This motivates the ensemble approach.
   - The previous methodology formulates the objective by scalarizing the distance objective and the limited mean (MDR) constraint. The reason why it worked poorly on maximum likelihood but worked well on kolmogorov distance is (i) log-likelihood and the difference in means are not homogeneous, which makes the scalarization hardly appropriate; (ii) maximum distance in CDF and the difference in means are more or less on the same scale. The new methodology solves d precisely given MDR, a, b, c on the fly, which enhances the goodness of fit to the data.
8. Bidirectional sequential fitting:
   - An extra recourse to smoothing TrB parameter transition along the MDR axis. Use optimized TrB parameters for the current MDR as initialization for the next MDR:
   - (i) Select the mean of the MDRs in data, e.g., 0.08. (ii) Fit the empirical cond PMF associated to 0.08. (iii) Use the optimized parameters as the initialization for fitting the PMF(0.08001), PMF(0.08002), ..., PMF(0.9999). (iv) Do the same for PMF(0.07999), PMF(0.07998), ..., PMF(0.00001).
   - Why bidirectional? — to start with a cond empirical PMF estimated (not extrapolated) from data.
   - Smoother transition + substantial speedup (25x) thanks to locality of the optima — need fewer iterations to converge to. Time cost: 3 seconds for fitting 18999 PMFs.
9. Discretization.
   - Was a little too ambitious. Wanted to find a discretization method that will keep P_0 intact.
   - But for TrB with low MDR and heavy tail, a large probability mass can be concentrated within the first delta (step on PMF support). It seems impossible to redistribute this mass without using the first support point (P_0) while bringing negligible changes to other parts of the distribution. It should be feasible if support size can be larger, e.g., 1024.
   - Stick to the 2019 discretization methodology. Improved and integrated it into the pipeline.

P_0 and means of empirical PMFs

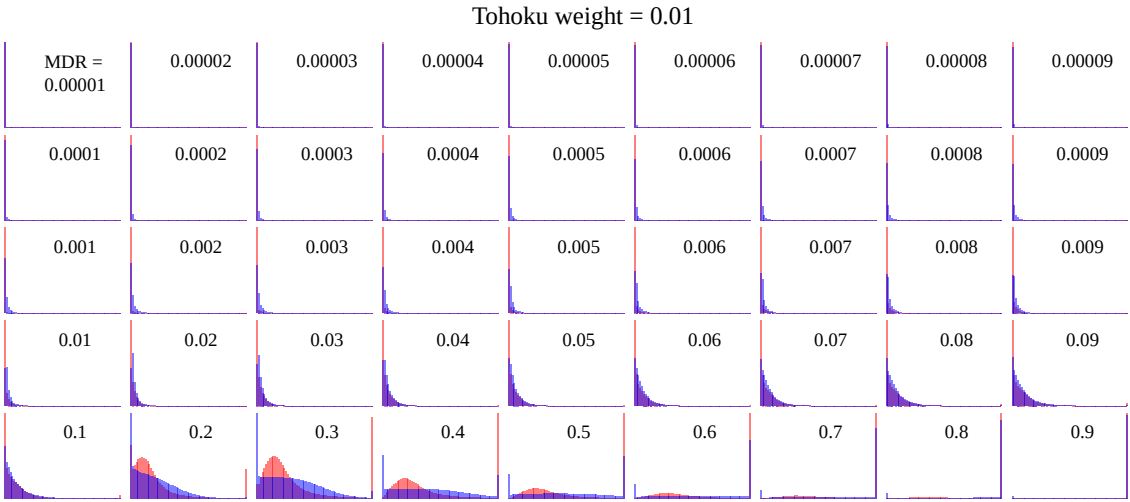


- Gray zone shows the data range.
- P_0 is the mean of an ensemble of size 30.
- Plot will be generated during pipeline execution to inform bias in the catastrophe model.

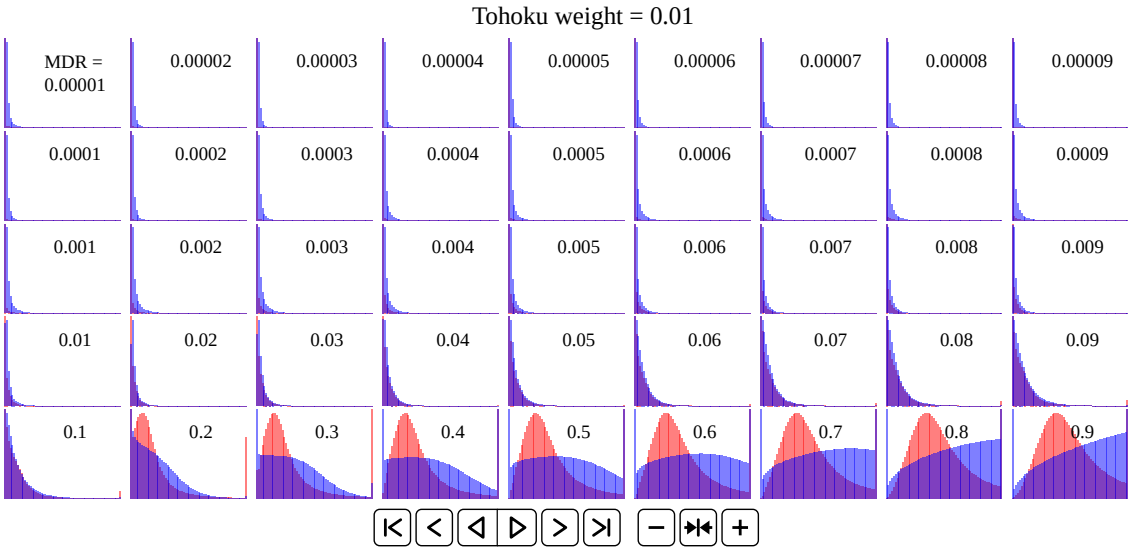
Fitted parameters



Bayesian update using Tohoku PMFs



Bayesian update using Tohoku PMFs, microscope on the main part



Bayesian update using Tohoku PMFs, microscope on the main part

Reasons for the persistent mismatch in the main parts for high weights and high MDRs:

1. The previous methodology extrapolates TrBs for high MDRs by increasing the scale parameter d of the same TrB, but it also adjusts $P_{\max}$ separately for meeting QA requirements — by scaling up $P_{\max}$ and scaling down the probabilities of the main part. Effectively, the final discretization is a combination of two separate models.
2. The two separate models are hard to be fitted by a single TrB model.
3. The overall difference is negligible due to the dominance of $P_{\max}$.

Future work

Test the pipeline on all available claims data to ensure its robustness.

Stress test the pipeline with garbage data.

1. Populate random uniform claim damage ratios.
2. Populate claim damage ratios that are far away from MDRs.
3. Populate claim damage ratios that are highly negatively correlated with MDRs.

Create an R package and document it like `keyALGs`.

Slides remade for brevity and clarity 20230713 (New Zealand earthquake)

Main steps

Assuming no deductibles and no incorporation of old distributions:

1. Import data: MDRs, damage ratios.
2. Model P_0.
3. Compute 18999 empirical PMFs.
4. Fit Transformed Beta (TrB) to the 18999 PMFs.
5. Discretization.

Import and order data

Import data: (MDRs, claim damage ratios).

1. Order data (rows) by MDR and claim damage ratio (CDR).
2. Reshuffle data using a user-defined random seed.
3. Order data by MDR.

If MDRs are not unique, different orders of rows can lead to different empirical distributions in bins.

Assume minimum bin size required = 5

	MDR	CDR		MDR	CDR	
MDR label for bin = 0.0525	0.05	0.01		0.05	0.01	
	0.05	0.02		0.05	0.02	CDRs in bin are different
	0.05	0.06		0.05	0.06	
	0.06	0.02		0.06	0.13	
	0.06	0.03		0.06	0.11	
	0.06	0.08		0.06	0.08	
	0.06	0.11		0.06	0.03	
	0.06	0.13		0.06	0.02	

Import and order data

Import data: (MDRs, claim damage ratios).

1. Order data (rows) by MDR and claim damage ratio (CDR).
2. Reshuffle data using a user-defined random seed.
3. Order data by MDR.

If MDRs are not unique, different orders of rows can lead to different empirical distributions in bins.

Even if CDRs are not sorted upon import, they could still be semi-sorted depending on how the data records were organized in databases. For example, the records could have been grouped by geographic regions / events / years, etc.

Assume minimum bin size required = 5

MDR label for bin = 0.0525

MDR	CDR	MDR	CDR
0.05	0.01	0.05	0.01
0.05	0.02	0.05	0.02
0.05	0.06	0.05	0.06
0.06	0.02	0.06	0.13
0.06	0.03	0.06	0.11
0.06	0.08	0.06	0.08
0.06	0.11	0.06	0.03
0.06	0.13	0.06	0.02

CDRs in bin are different

Import and order data

Import data: (MDRs, claim damage ratios).

1. Order data (rows) by MDR and claim damage ratio (CDR).
2. Reshuffle data using a user-defined random seed.
3. Order data by MDR.

If MDRs are not unique, different orders of rows can lead to different empirical distributions in bins.

Even if CDRs are not sorted upon import, they could still be semi-sorted depending on how the data records were organized in databases. For example, the records could have been grouped by geographic regions / events / years, etc.

The "order + shuffle + order" procedure removes any potential bias and guarantees uniqueness of empirical distributions in bins, i.e. re-producibility.

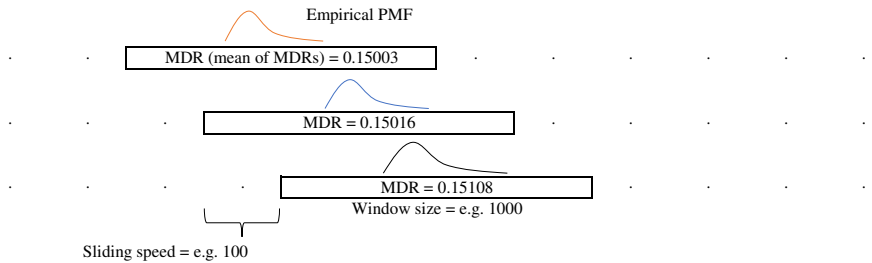
Assume minimum bin size required = 5

MDR label for bin = 0.0525

MDR	CDR	MDR	CDR
0.05	0.01	0.05	0.01
0.05	0.02	0.05	0.02
0.05	0.06	0.05	0.06
0.06	0.02	0.06	0.13
0.06	0.03	0.06	0.11
0.06	0.08	0.06	0.08
0.06	0.11	0.06	0.03
0.06	0.13	0.06	0.02

CDRs in bin are different

Sliding window and sliding speed



Overlapped windows $\Longrightarrow$ correlated samples $\Longrightarrow$ close empirical PMFs in shape and scale $\Longrightarrow$ smoother transition of TrBs along the MDR axis.

Sliding windows will also be used to model P_0.

Model P_0

Set sliding window size and speed.

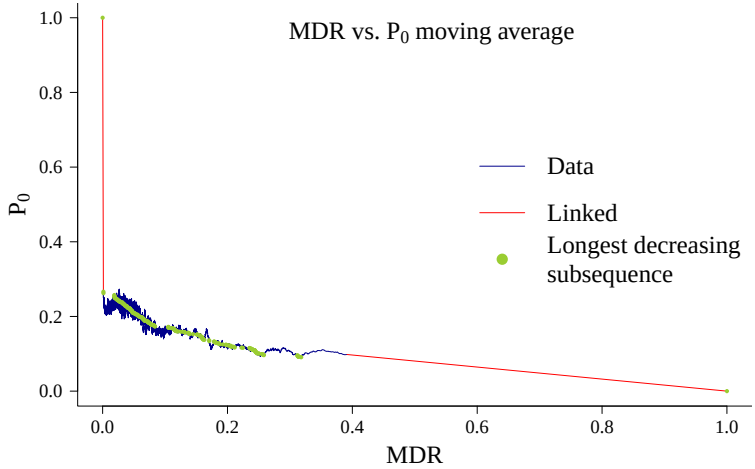
Compute P_0 and mean of MDRs in each window.

Constraint: P_0 is a decreasing function of MDR.

- $P_0(\text{MDR} = 0) = 1$.
- $P_0(\text{MDR} = 1) = 0$.

Compute the longest decreasing subsequence.

- $O(N \log N)$ [REF].



Model P_0

Set sliding window size and speed.

Compute P_0 and mean of MDRs in each window.

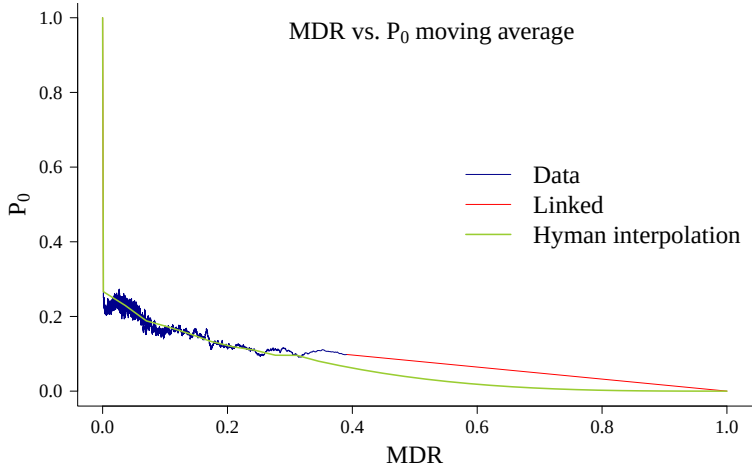
Constraint: P_0 is a decreasing function of MDR.

- $P_0(\text{MDR} = 0) = 1$.
- $P_0(\text{MDR} = 1) = 0$.

Compute the longest decreasing subsequence.

- $O(N \log N)$ [REF].

Monotone cubic interpolation of the subsequence [REF].



Model P_0, robustification (not necessary but good to have and cheap to run)

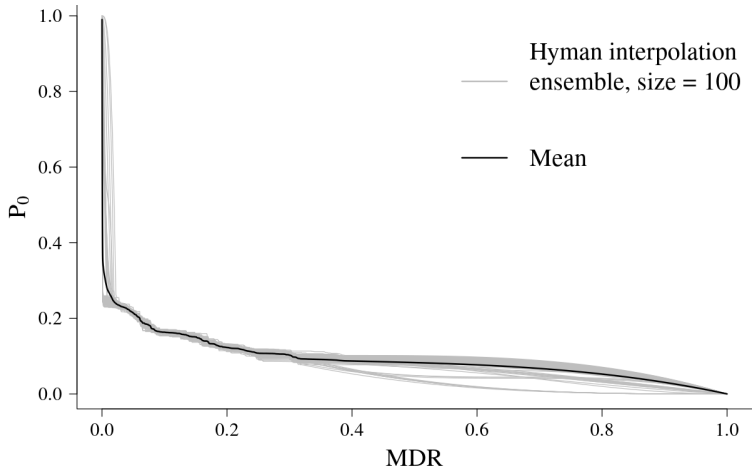
Sample 100 random subsets of data. Subset size = e.g. 2/3 of the full data size.

Compute the Hyman model in each subset.

Ensemble mean is used as the final estimate.

Note:

- To prevent overfitting, the main part i.e. $P(\text{DR} > 0)$ can also be modeled using ensemble. This would produce 100 sets of TrB parameters. The final PMF table would be a mixture of the discretizations of the 100 sets of TrBs. We ignore this approach for now.



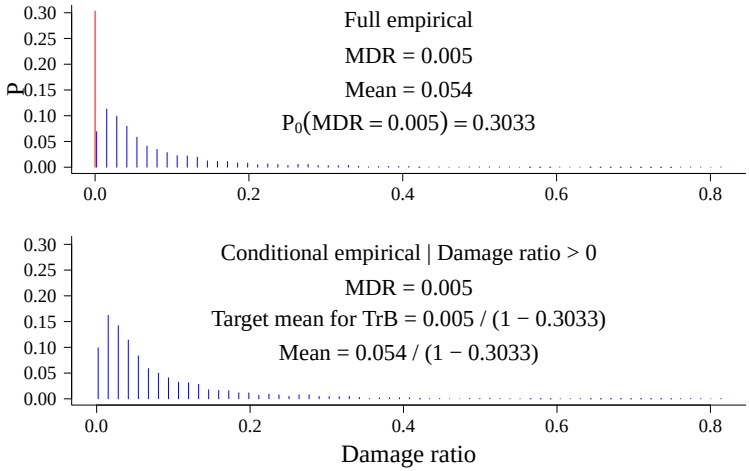
Model the main part $P(\text{DR} > 0)$

Remove data where DR $= 0$.

Set sliding window and speed.

Compute MDR mean and conditional empirical PMF in each window.

- P_0 (red) is given.
- Empirical PMF is computed using regriding.



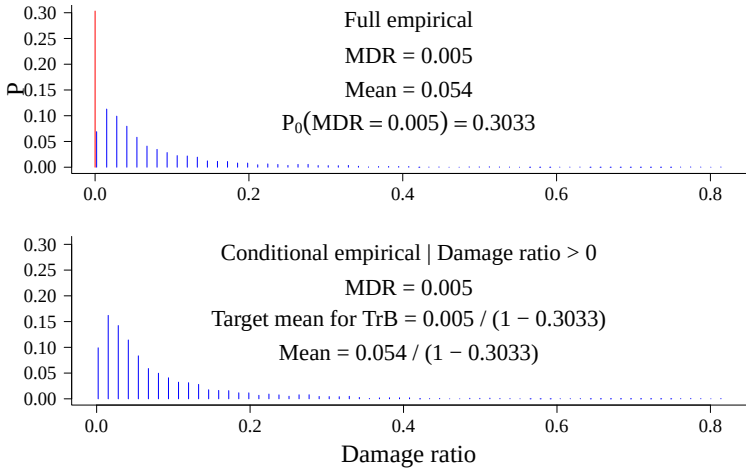
Model the main part $P(\text{DR} > 0)$

Remove data where DR $= 0$.

Set sliding window and speed.

Compute MDR mean and conditional empirical PMF in each window.

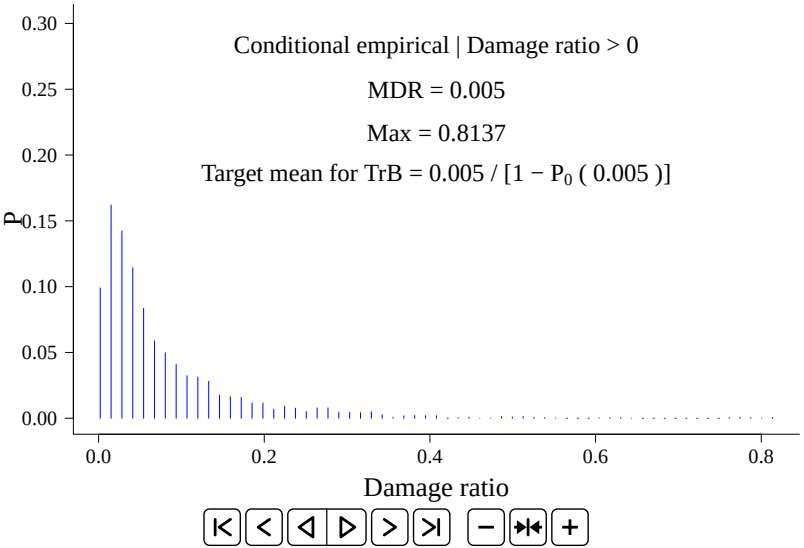
- P_0 (red) is given.
- Empirical PMF is computed using regriding.
- For prescribed MDR that is not windowed, mix the neighboring PMFs (explained previously).
- For prescribed MDR that is out of data range, extrapolate.



Extrapolation below minimum MDR in data

Example assumes 0.005 the lowest MDR in data.

Downscale the support of PMF for MDR $= 0.005$ linear to the prescribed MDRs.

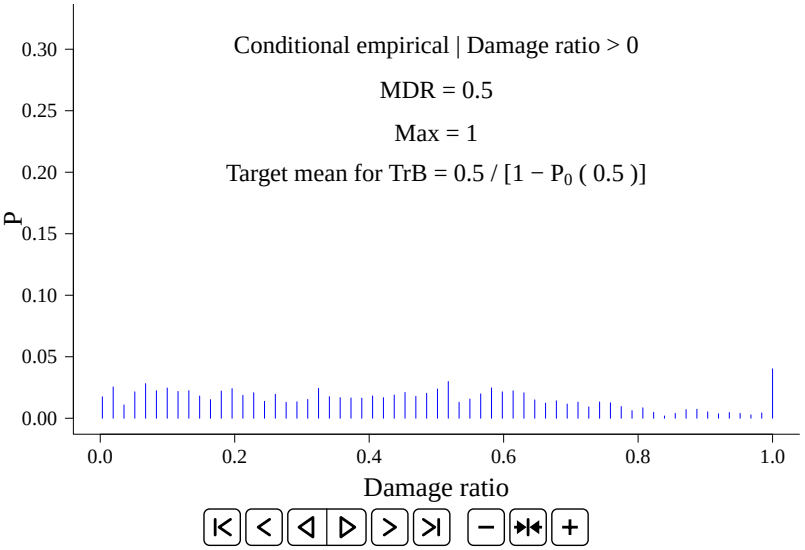


Extrapolation above maximum MDR in data

Example assumes 0.5 the highest MDR in data.

Upscale the support of PMF for MDR $= 0.5$ linear to the prescribed MDRs.

Upper bound the PMF's support by 1 and load the truncated tail on $P_{\max}$.



Fit TrB

Denote the incomplete beta function by $\beta(u, v; y) = \int_0^y t^{u-1}(1-t)^{v-1}dt$. For TrB random variable X,

$$\begin{aligned}\mu_{\text{lm1}} &= \mathbb{E}[\min(X, 1)] \\ &= \frac{d \cdot \Gamma\left(\frac{b+1}{c}\right)\Gamma\left(\frac{a-1}{c}\right)}{\Gamma\left(\frac{a}{c}\right)\Gamma\left(\frac{b}{c}\right)} \cdot \beta\left(\frac{b+1}{c}, \frac{a-1}{c}; \frac{1}{d^c+1}\right) + 1 - \beta\left(\frac{b}{c}, \frac{a}{c}; \frac{1}{d^c+1}\right).\end{aligned}$$

Fit TrB

Denote the incomplete beta function by $\beta(u, v; y) = \int_0^y t^{u-1}(1-t)^{v-1}dt$. For TrB random variable X,

$$\begin{aligned}\mu_{\text{lm1}} &= \mathbb{E}[\min(X, 1)] \\ &= \frac{d \cdot \Gamma\left(\frac{b+1}{c}\right)\Gamma\left(\frac{a-1}{c}\right)}{\Gamma\left(\frac{a}{c}\right)\Gamma\left(\frac{b}{c}\right)} \cdot \beta\left(\frac{b+1}{c}, \frac{a-1}{c}; \frac{1}{d^c+1}\right) + 1 - \beta\left(\frac{b}{c}, \frac{a}{c}; \frac{1}{d^c+1}\right).\end{aligned}$$

Interestingly,

$$\frac{\partial \mu_{\text{lm1}}}{\partial d} = \frac{\Gamma\left(\frac{b+1}{c}\right)\Gamma\left(\frac{a-1}{c}\right)}{\Gamma\left(\frac{a}{c}\right)\Gamma\left(\frac{b}{c}\right)} \cdot \beta\left(\frac{b+1}{c}, \frac{a-1}{c}; \frac{1}{d^c+1}\right) > 0,$$

so μ_{lm1} is an increasing function of d and the derivative can be obtained almost free of charge after computing μ_{lm1}.

Fit TrB

Denote the incomplete beta function by $\beta(u, v; y) = \int_0^y t^{u-1}(1-t)^{v-1}dt$. For TrB random variable X,

$$\begin{aligned}\mu_{\text{lm1}} &= \mathbb{E}[\min(X, 1)] \\ &= \frac{d \cdot \Gamma\left(\frac{b+1}{c}\right)\Gamma\left(\frac{a-1}{c}\right)}{\Gamma\left(\frac{a}{c}\right)\Gamma\left(\frac{b}{c}\right)} \cdot \beta\left(\frac{b+1}{c}, \frac{a-1}{c}; \frac{1}{d^c+1}\right) + 1 - \beta\left(\frac{b}{c}, \frac{a}{c}; \frac{1}{d^c+1}\right).\end{aligned}$$

Interestingly,

$$\frac{\partial \mu_{\text{lm1}}}{\partial d} = \frac{\Gamma\left(\frac{b+1}{c}\right)\Gamma\left(\frac{a-1}{c}\right)}{\Gamma\left(\frac{a}{c}\right)\Gamma\left(\frac{b}{c}\right)} \cdot \beta\left(\frac{b+1}{c}, \frac{a-1}{c}; \frac{1}{d^c+1}\right) > 0,$$

so μ_{lm1} is an increasing function of d and the derivative can be obtained almost free of charge after computing μ_{lm1}.

Because the target mean for $\min(X, 1)$ is also constrained by:

$$\mu^*_{\text{lm1}} = \frac{\text{MDR}}{1 - P_0(\text{MDR})},$$

we can quickly solve d given $a, b, c, \mu^*_{\text{lm1}}$ using Newton's method.

Fit TrB

Denote the incomplete beta function by $\beta(u, v; y) = \int_0^y t^{u-1}(1-t)^{v-1}dt$. For TrB random variable X,

$$\begin{aligned}\mu_{\text{lm1}} &= \mathbb{E}[\min(X, 1)] \\ &= \frac{d \cdot \Gamma\left(\frac{b+1}{c}\right)\Gamma\left(\frac{a-1}{c}\right)}{\Gamma\left(\frac{a}{c}\right)\Gamma\left(\frac{b}{c}\right)} \cdot \beta\left(\frac{b+1}{c}, \frac{a-1}{c}; \frac{1}{d^c+1}\right) + 1 - \beta\left(\frac{b}{c}, \frac{a}{c}; \frac{1}{d^c+1}\right).\end{aligned}$$

Interestingly,

$$\frac{\partial \mu_{\text{lm1}}}{\partial d} = \frac{\Gamma\left(\frac{b+1}{c}\right)\Gamma\left(\frac{a-1}{c}\right)}{\Gamma\left(\frac{a}{c}\right)\Gamma\left(\frac{b}{c}\right)} \cdot \beta\left(\frac{b+1}{c}, \frac{a-1}{c}; \frac{1}{d^c+1}\right) > 0,$$

so μ_{lm1} is an increasing function of d and the derivative can be obtained almost free of charge after computing μ_{lm1}.

Because the target mean for $\min(X, 1)$ is also constrained by:

$$\mu^*_{\text{lm1}} = \frac{\text{MDR}}{1 - P_0(\text{MDR})},$$

we can quickly solve d given $a, b, c, \mu^*_{\text{lm1}}$ using Newton's method.

The mean constraint removes one degree of freedom. Fitting TrB is only 3-d optimization for a, b, c.

Fit TrB

Algorithm I: Negative log-likelihood $\mathcal{L}(a, b, c, d, x[], p[])$

INPUT: TrB parameters a, b, c, d; Empirical PMF support $x[N]$ and probabilities $p[N]$; TrB PDF f and CDF F.

OUTPUT: Negative log-likelihood.

1. $l \leftarrow 0$
2. **for** $i = 1$ to $N - 1$:
   $l \mathrel{+}= -p[i] \log f(a, b, c, d; x[i])$
3. **if** $x[N] < 1$: $l \mathrel{+}= -p[i] \log f(a, b, c, d; x[i])$
   else: $l \mathrel{+}= -p[i] \log(1 - F(a, b, c, d; 1))$
4. **return** l

Negative log-likelihood of weighted data (PMF) against a continuous PDF is equivalent to cross entropy.

Fit TrB

Algorithm I: Negative log-likelihood $\mathcal{L}(a, b, c, d, x[], p[])$

INPUT: TrB parameters a, b, c, d; Empirical PMF support $x[N]$ and probabilities $p[N]$; TrB PDF f and CDF F.

OUTPUT: Negative log-likelihood.

1. $l \leftarrow 0$
2. **for** $i = 1$ to $N - 1$:
   $l \mathrel{+}= -p[i] \log f(a, b, c, d; x[i])$
3. **if** $x[N] < 1$: $l \mathrel{+}= -p[i] \log f(a, b, c, d; x[i])$
   else: $l \mathrel{+}= -p[i] \log(1 - F(a, b, c, d; 1))$
4. **return** l

Negative log-likelihood of weighted data (PMF) against a continuous PDF is equivalent to cross entropy.

We unify gradient computation by using finite difference (FD):

1. The analytical form of $\partial\mathcal{L}/\partial(a, b, c)$ is not simple enough to bring meaningful speedup.
2. Other distance measures have costlier analytical gradient forms.
3. FD can be more numerically stable.

Algorithm II: Objective function $\mathcal{O}$ and gradient $\nabla\mathcal{O}$

INPUT: Empirical PMF's support $x[N]$, probabilities $p[N]$, target limited mean μ^*_{lm1}; TrB parameters a, b, c; Newton's 1-d root finder $\mathcal{R}$; Distance function $\mathcal{L}$.

OUTPUT: Object function value and gradient with respective to (a, b, c).

1. $d \leftarrow \mathcal{R}\left(a, b, c, \mu^*_{\mathsf{lm1}}\right)$
2. **return** $\mathcal{L}(a, b, c, d, x[N], p[N]), \left(\frac{\partial\mathcal{L}}{\partial a}, \frac{\partial\mathcal{L}}{\partial b}, \frac{\partial\mathcal{L}}{\partial c}\right)$

Root finder $\mathcal{R}$ invokes bisection contingent on Newton's divergence.

Fit TrB

Algorithm I: Negative log-likelihood $\mathcal{L}(a, b, c, d, x[], p[])$

INPUT: TrB parameters a, b, c, d; Empirical PMF support $x[N]$ and probabilities $p[N]$; TrB PDF f and CDF F.

OUTPUT: Negative log-likelihood.

1. $l \leftarrow 0$
2. **for** $i = 1$ to $N - 1$:
   $l \mathrel{+}= -p[i] \log f(a, b, c, d; x[i])$
3. **if** $x[N] < 1$: $l \mathrel{+}= -p[i] \log f(a, b, c, d; x[i])$
   else: $l \mathrel{+}= -p[i] \log(1 - F(a, b, c, d; 1))$
4. **return** l

Negative log-likelihood of weighted data (PMF) against a continuous PDF is equivalent to cross entropy.

We unify gradient computation by using finite difference (FD):

1. The analytical form of $\partial\mathcal{L}/\partial(a, b, c)$ is not simple enough to bring meaningful speedup.
2. Other distance measures have costlier analytical gradient forms.
3. FD can be more numerically stable.

Algorithm II: Objective function $\mathcal{O}$ and gradient $\nabla\mathcal{O}$

INPUT: Empirical PMF's support $x[N]$, probabilities $p[N]$, target limited mean μ^*_{lm1}; TrB parameters a, b, c; Newton's 1-d root finder $\mathcal{R}$; Distance function $\mathcal{L}$.

OUTPUT: Object function value and gradient with respective to (a, b, c).

1. $d \leftarrow \mathcal{R}\left(a, b, c, \mu^*_{\text{lm1}}\right)$
2. **return** $\mathcal{L}(a, b, c, d, x[N], p[N]), \left(\frac{\partial\mathcal{L}}{\partial a}, \frac{\partial\mathcal{L}}{\partial b}, \frac{\partial\mathcal{L}}{\partial c}\right)$

Root finder $\mathcal{R}$ invokes bisection contingent on Newton's divergence.

Algorithm III: Solve a, b, c, d

INPUT: Empirical PMF's support $x[N]$, probabilities $p[N]$, and target limited mean μ^*_{lm1}; Quasi-Newton minimizer $\mathcal{Q}$, e.g., L-BFGS-B; all previously defined functions.

OUTPUT: a, b, c, d

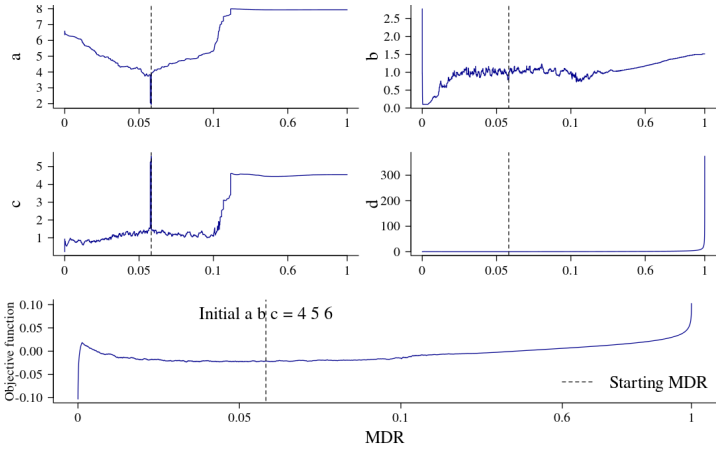
1. $a, b, c \leftarrow \mathcal{Q}(\mathcal{O}, \nabla\mathcal{O})$
2. $d \leftarrow \mathcal{R}\left(a, b, c, \mu^*_{\text{lm1}}\right)$
3. **return** (a, b, c, d)

Bidirectional sequential fitting

Bidirectional sequential fitting:

1. Select an MDR from data, e.g. median(MDR) = 0.058.
2. Fit TrB to the empirical PMF associated with the MDR.
3. Go to the next (previous) MDR, use the current optimized parameters as initialization, and rerun Step 2.

The motivation is to promote the locality of optima. The result (a, b, c, d) are still not perfectly smooth functions of MDR, which is expected.



Bidirectional sequential fitting

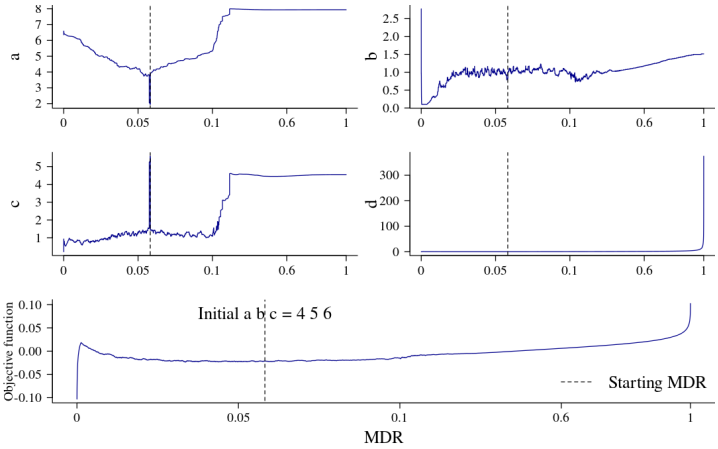
Bidirectional sequential fitting:

1. Select an MDR from data, e.g. median(MDR) = 0.058.
2. Fit TrB to the empirical PMF associated with the MDR.
3. Go to the next (previous) MDR, use the current optimized parameters as initialization, and rerun Step 2.

The motivation is to promote the locality of optima. The result (a, b, c, d) are still not perfectly smooth functions of MDR, which is expected.

The drop and surge in the middle of a and c correspond to about 300 MDRs surrounding 0.058 — the starting MDR. The behavior is still being investigated, but it is probably because cumulative change in the empirical PMF becomes large enough to push the optimizer towards a far-away local minimum — even if this local minimum is minimally better. The diagnose is reasonable but still superficial since it does not explain why such behavior only occurs near the starting MDR: if we change the starting MDR to, e.g., 0.1, the drop and surge will appear around MDR=0.1.

The drop and surge do not translate to similar behaviors in TrB density or in the objective function.



Bidirectional sequential fitting (update)

It has been confirmed that the drop and surge are due to "bad" initialization at the starting MDR.

The objective function has almost no change at all given quite different parameters.

This implies numerous local optima for the same goodness of fit, and suggests that the distribution model might be overparameterized. Three-parameter members in the TrB family, i.e. Burr, Generalized Pareto, Inverse Burr, could be worth a trial **in the distant future**.

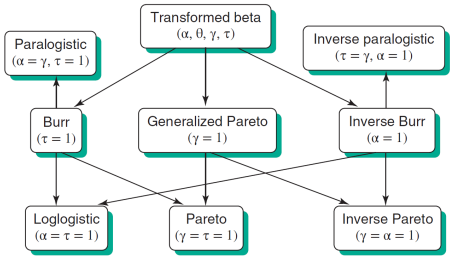
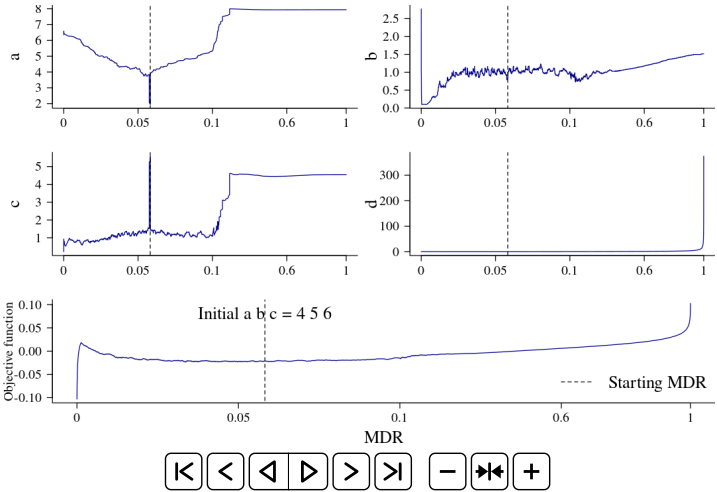


Figure 5.2 The transformed beta family.

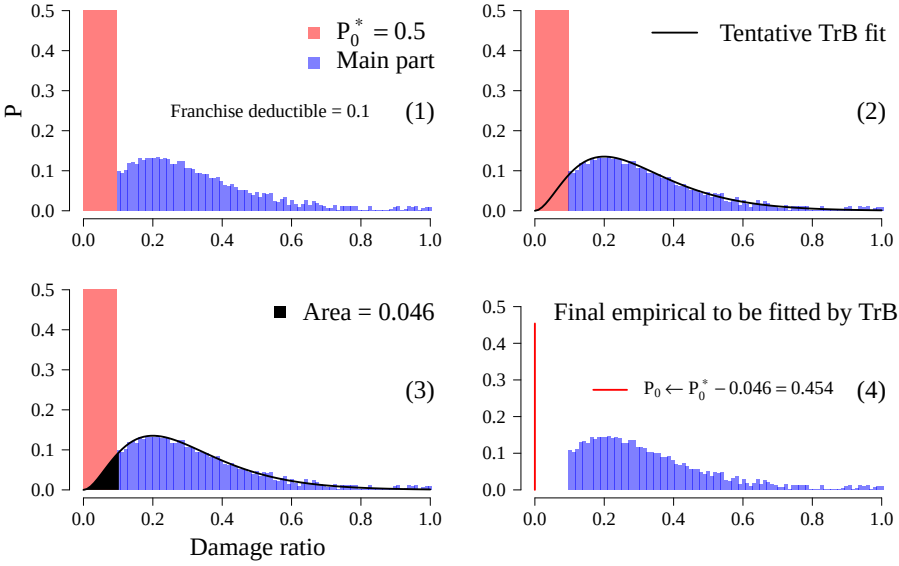


Bayesian update

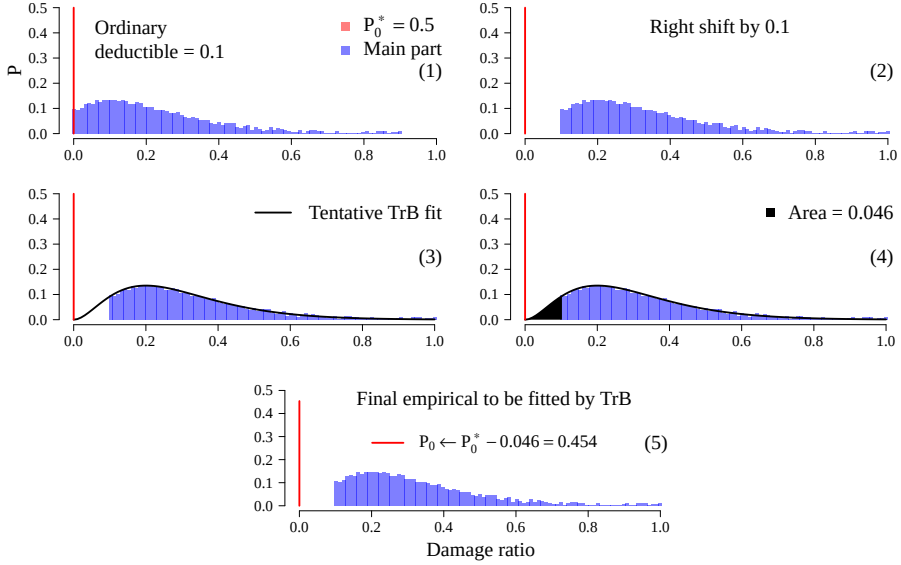
Assuming no deductibles:

1. Import data: (MDRs, claim damage ratios).
   - Also import the old PMF table, and set the update weight w.
2. Model P_0.
   - $P_0 \leftarrow (1-w)P_0 + wP_0^{\text{old}}$.
3. Compute 18999 conditional empirical PMFs.
   - $P_{\text{DR}>0} \leftarrow (1-w)P_{\text{DR}>0} + wP_{\text{DR}>0}^{\text{old}}$.
4. Fit Transformed Beta (TrB) to the 18999 PMFs.
5. Discretization.

Given franchise deductible



Given ordinary deductible



Raw discretization

1. Set tentative quantile q e.g. 0.999. Given an MDR, let $\max(\text{MDR})$ be the max of PMF's support, let F^{-1}_{MDR} be the inverse TrB CDF:
   - $\max(\text{MDR}) \leftarrow \min\left(1, F^{-1}_{\text{MDR}}(q)\right)$.
   - If $\max(\text{MDR})$ is not a nondecreasing sequence (rarely happens), compute the longest nondecreasing subsequence and do Hyman interpolation.

Raw discretization

1. Set tentative quantile q e.g. 0.999. Given an MDR, let $\max(\text{MDR})$ be the max of PMF's support, let F^{-1}_{MDR} be the inverse TrB CDF:
   - $\max(\text{MDR}) \leftarrow \min\left(1, F^{-1}_{\text{MDR}}(q)\right)$.
   - If $\max(\text{MDR})$ is not a nondecreasing sequence (rarely happens), compute the longest nondecreasing subsequence and do Hyman interpolation.
2. For each MDR,
   2.1 Set a fine support of e.g. 2000 points, i.e. $\{\max/2000, 2 \cdot \max/2000, \ldots, \max\}$. Discretize TrB onto the support via central differencing TrB CDF.
      - If $\max = 1$: $P_{\max} \leftarrow 1 - F(1 - \max/4000)$.
   2.2 Renormalize the fine discretization such that the sum of probabilities equals $1 - P_0$. Prepend $(0, P_0)$ to the fine discretization.

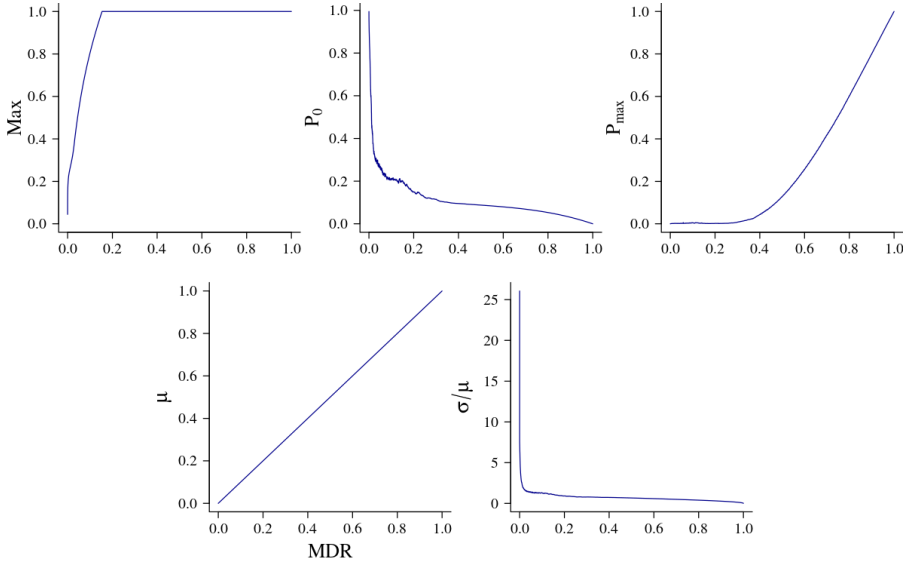
Raw discretization

1. Set tentative quantile q e.g. 0.999. Given an MDR, let $\max(\text{MDR})$ be the max of PMF's support, let F^{-1}_{MDR} be the inverse TrB CDF:
   - $\max(\text{MDR}) \leftarrow \min\left(1, F^{-1}_{\text{MDR}}(q)\right)$.
   - If $\max(\text{MDR})$ is not a nondecreasing sequence (rarely happens), compute the longest nondecreasing subsequence and do Hyman interpolation.
2. For each MDR,
   - 2.1 Set a fine support of e.g. 2000 points, i.e. $\{\max/2000, 2 \cdot \max/2000, \ldots, \max\}$. Discretize TrB onto the support via central differencing TrB CDF.
     - If $\max = 1$: $P_{\max} \leftarrow 1 - F(1 - \max/4000)$.
   - 2.2 Renormalize the fine discretization such that the sum of probabilities equals $1 - P_0$. Prepend $(0, P_0)$ to the fine discretization.
   - 2.3 Regrid the PMF onto the final 42/64-point support.

Raw discretization

1. Set tentative quantile q e.g. 0.999. Given an MDR, let $\max(\text{MDR})$ be the max of PMF's support, let F^{-1}_{MDR} be the inverse TrB CDF:
   - $\max(\text{MDR}) \leftarrow \min\left(1, F^{-1}_{\text{MDR}}(q)\right)$.
   - If $\max(\text{MDR})$ is not a nondecreasing sequence (rarely happens), compute the longest nondecreasing subsequence and do Hyman interpolation.
2. For each MDR,
   - 2.1 Set a fine support of e.g. 2000 points, i.e. $\{\max/2000, 2 \cdot \max/2000, \ldots, \max\}$. Discretize TrB onto the support via central differencing TrB CDF.
     - If $\max = 1$: $P_{\max} \leftarrow 1 - F(1 - \max/4000)$.
   - 2.2 Renormalize the fine discretization such that the sum of probabilities equals $1 - P_0$. Prepend $(0, P_0)$ to the fine discretization.
   - 2.3 Regrid the PMF onto the final 42/64-point support.
   - 2.4 Scale up/down P_0 while scaling down/up all the main probabilities together to eliminate small error between MDR and PMF's mean.

Raw discretization



Tune discretization

Given a PMF table of size 19001 $\times$ $(N+2)$, $N \in \{42, 64\}$:

1. Impose monotonicity on P_0.

   1.1 Compute the longest nonincreasing subsequence of P_0. Retrieve all the PMFs associated to the subsequence.

   1.2 Let PMF_i and PMF_j be any two neighboring PMFs in the PMF sequence. Denote their MDRs by MDR_i and MDR_j, $i < j$.

   1.3 If there should have been a prescribed MDR^* between MDR_i and MDR_j:

   $$w \leftarrow (\text{MDR}^* - \text{MDR}_i)/(\text{MDR}_j - \text{MDR}_i)$$
   $$\text{PMF}^* \leftarrow (1-w)\text{PMF}_i + w\text{PMF}_j$$

   1.4 Insert PMF^* to the PMF table. If the table size has not reached 19001, return to Step 1.2.

2. Impose monotonicity on $P_{\max}$ in a similar way, which will not violate P_0's monotonicity.

3. By now, monotonicity in maxes might have been violated. Re-impose monotonicity in maxes:

   3.1 Compute the longest nondecreasing subsequence of maxes. Retrieve all the PMFs associated with the subsequence.

3.

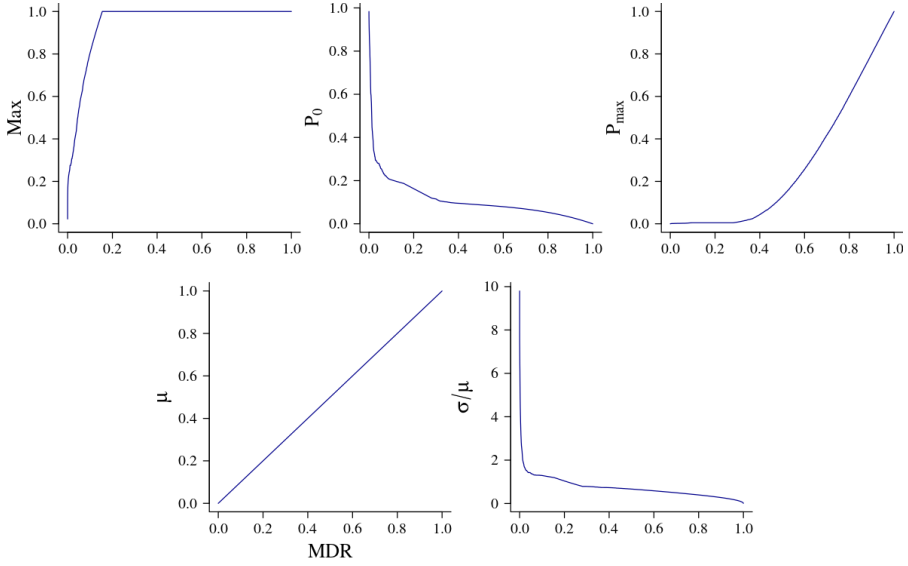
   3.2 Let PMF_i and PMF_j be any two neighboring PMFs in the sequence. Let MDR_i, MDR_j, $\max_i$, $\max_j$, $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, $\boldsymbol{p}_i$, $\boldsymbol{p}_j$, be their MDRs, maxes, supports and probability vectors respectively. Additionally, let $\boldsymbol{x}(\max_k)$

   3.3 If there should have been a prescribed MDR^* between MDR_i and MDR_j, do:

   $$\max^* \leftarrow (\max_i + \max_j)/2$$
   $$\boldsymbol{x}^* \leftarrow \left(0, \frac{\max^*}{N-1}, \frac{2\max^*}{N-1}, \ldots, \max^*\right)$$
   $$m_i \leftarrow \boldsymbol{x}^* \cdot \boldsymbol{p}_i$$
   $$m_j \leftarrow \boldsymbol{x}^* \cdot \boldsymbol{p}_j$$
   $$w \leftarrow (\text{MDR}^* - m_i)/(m_j - m_i)$$
   $$\boldsymbol{p}^* \leftarrow (1-w)\boldsymbol{p}_i + w\boldsymbol{p}_j$$

   3.4 Insert $\boldsymbol{p}^*$ into the PMF table. If the table size has not reached 19001, return to Step 3.2.

Tuned discretization



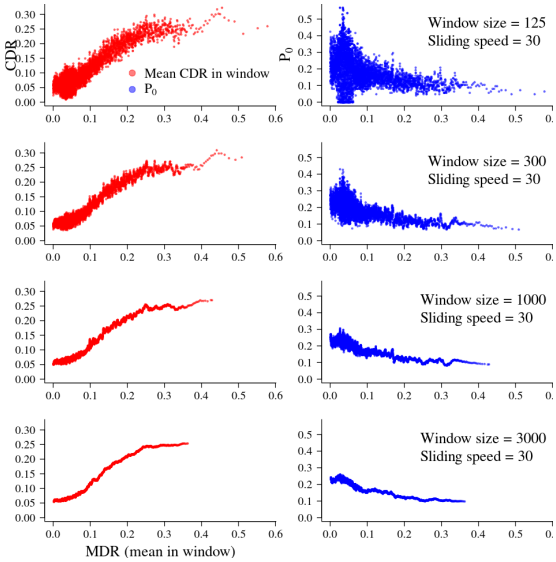
Sliding window sizes

Larger windows $\Longrightarrow$ smoother trends.

Larger windows $\Longrightarrow$ higher starting mean MDR, lower ending mean MDR.

Window size too large $\Longrightarrow$ oversmoothing.

Window size too small $\Longrightarrow$ less credible empirical PMFs in windows.



Sliding window sizes

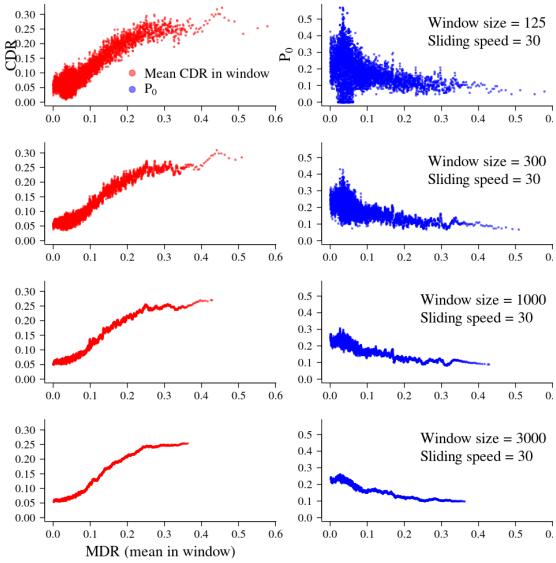
Larger windows $\Longrightarrow$ smoother trends.

Larger windows $\Longrightarrow$ higher starting mean MDR, lower ending mean MDR.

Window size too large $\Longrightarrow$ oversmoothing.

Window size too small $\Longrightarrow$ less credible empirical PMFs in windows.

- Empirical PMF has 64 points, thus window size ≥ 200 is recommended, aka $\sim$3 points on average for one probability bin.
- In principle, fitting distributions does not require inferring "empirical PMF" from data first. However,



Sliding window sizes

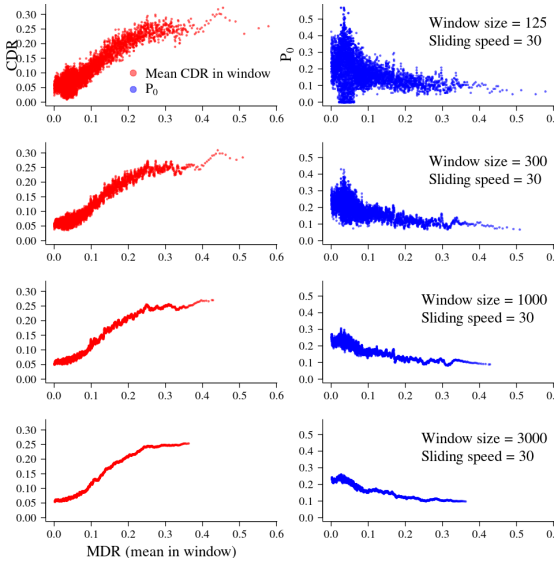
Larger windows $\Longrightarrow$ smoother trends.

Larger windows $\Longrightarrow$ higher starting mean MDR, lower ending mean MDR.

Window size too large $\Longrightarrow$ oversmoothing.

Window size too small $\Longrightarrow$ less credible empirical PMFs in windows.

- Empirical PMF has 64 points, thus window size ≥ 200 is recommended, aka $\sim$3 points on average for one probability bin.
- In principle, fitting distributions does not require inferring "empirical PMF" from data first. However,
  - Doing so is necessary for Bayesian update because old distributions are characterized by PMFs.
  - Empirical PMFs are needed for visualization of the goodness of fit.
  - Fitting to the empirical PMFs is direct — what you see is what you fit.



Sliding window sizes

Larger windows $\Longrightarrow$ smoother trends.

Larger windows $\Longrightarrow$ higher starting mean MDR, lower ending mean MDR.

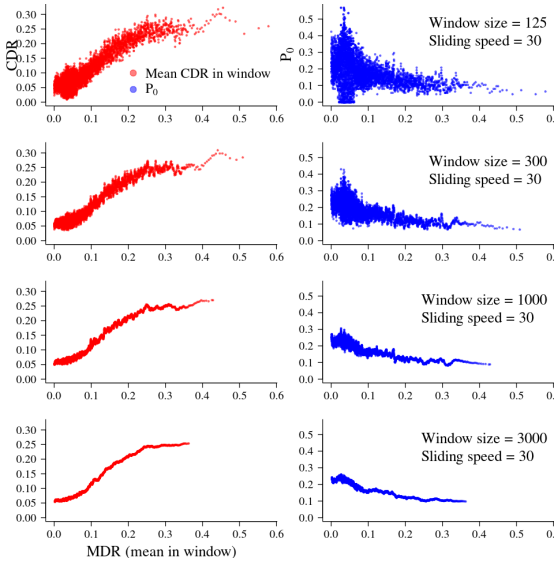
Window size too large $\Longrightarrow$ oversmoothing.

Window size too small $\Longrightarrow$ less credible empirical PMFs in windows.

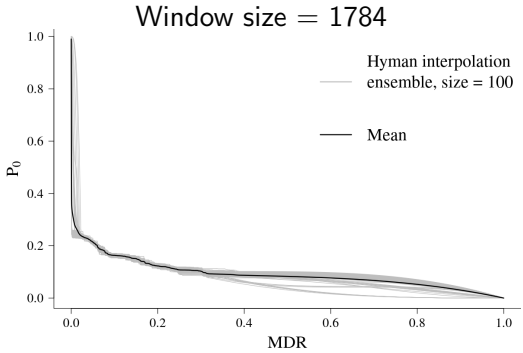
- Empirical PMF has 64 points, thus window size ≥ 200 is recommended, aka $\sim$3 points on average for one probability bin.
- In principle, fitting distributions does not require inferring "empirical PMF" from data first. However,
  - Doing so is necessary for Bayesian update because old distributions are characterized by PMFs.
  - Empirical PMFs are needed for visualization of the goodness of fit.
  - Fitting to the empirical PMFs is direct — what you see is what you fit.

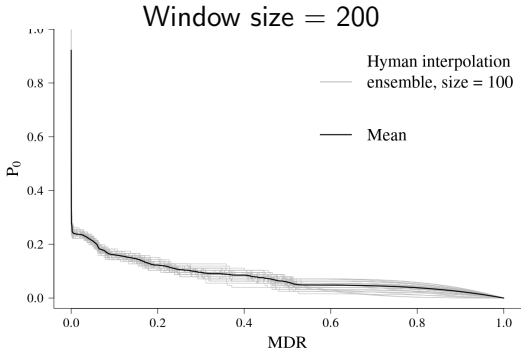
It is recommended to initialize different window sizes, run through the computing pipeline, compare and contrast the results before making decisions.

- Scoring the output distributions (explained in next slides) over claims data could be a useful criterion.



Sliding window size 1784 vs. 200, P_0s



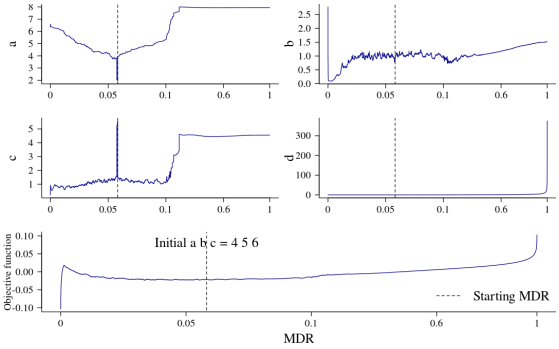


Smaller windows lead to less smooth interpolations.

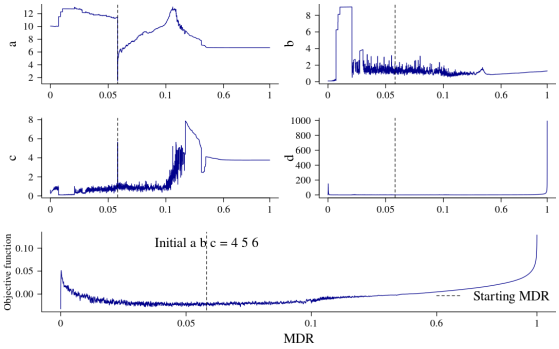
Given smaller windows, ensemble appears more necessary.

Sliding window size 1784 vs. 200, fitted parameters

Window size = 1784



Window size = 200



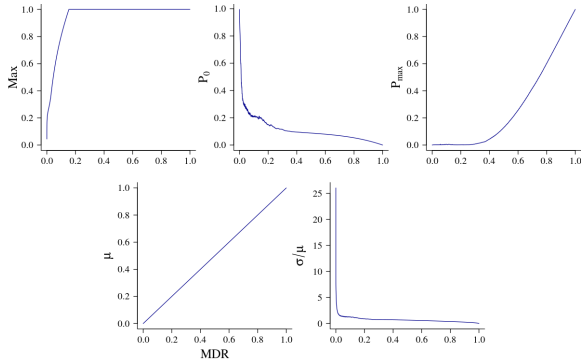
Smaller windows lead to rougher transitions in TrB parameters along the MDR axis.

Different window sizes result in different parameters.

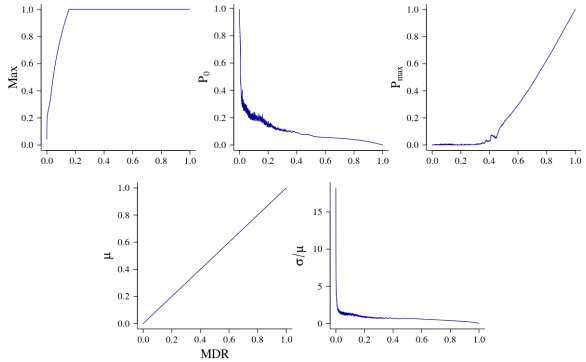
Objective function values however are similar given different window sizes.

Sliding window size 1784 vs. 200, raw discretization

Window size = 1784



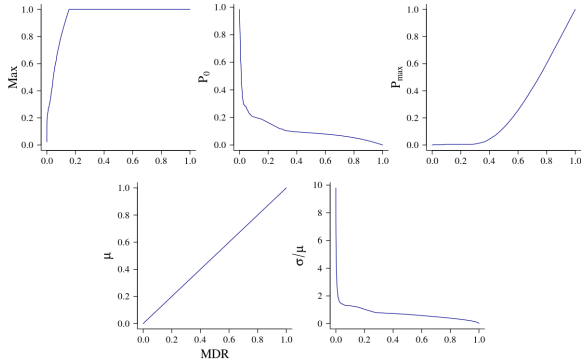
Window size = 200



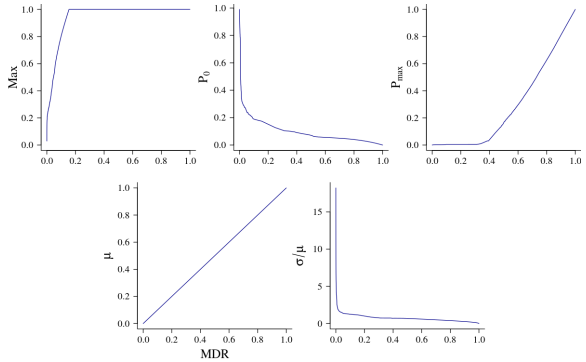
Smaller windows lead to fuzzier P_0, $P_{\max}$, μ/σ functions of MDR. More corrections will be taken for imposing monotonicities.

Sliding window size 1784 vs. 200, tuned discretization

Window size = 1784



Window size = 200

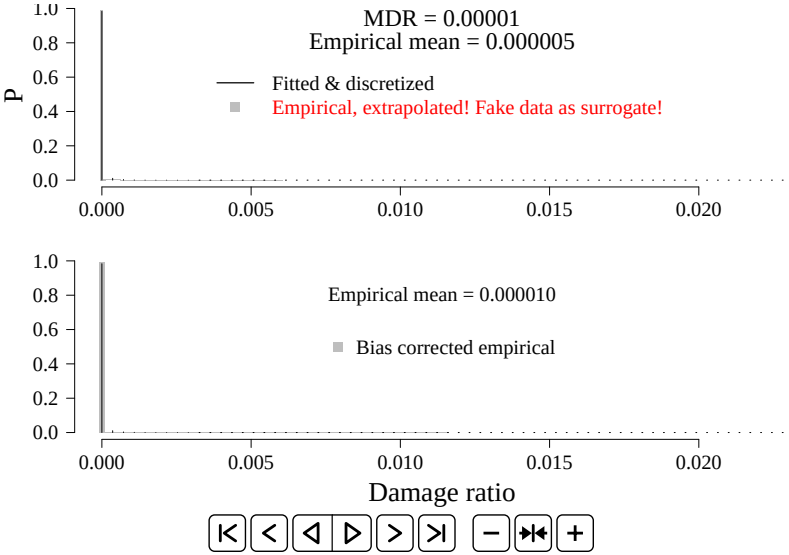


P_0s, $P_{\max}$s are still close given different window sizes.

μ/σs are close in most of the MDR range, but largely different when MDR is small. PMF with small MDR is dominated by P_0. Small perturbance in P_0 amplifies the difference in σ and thus μ/σ.

Sliding window size 1784, fitted vs. empirical

Plots are associated with 45 MDRs $\in$ { 1e-5, 2e-5, . . ., 1e-4, 2e-4, . . ., 1e-3, 2e-3, . . ., 1e-2, 2e-2, . . ., 1e-1, 2e-1, . . ., 9e-1 }

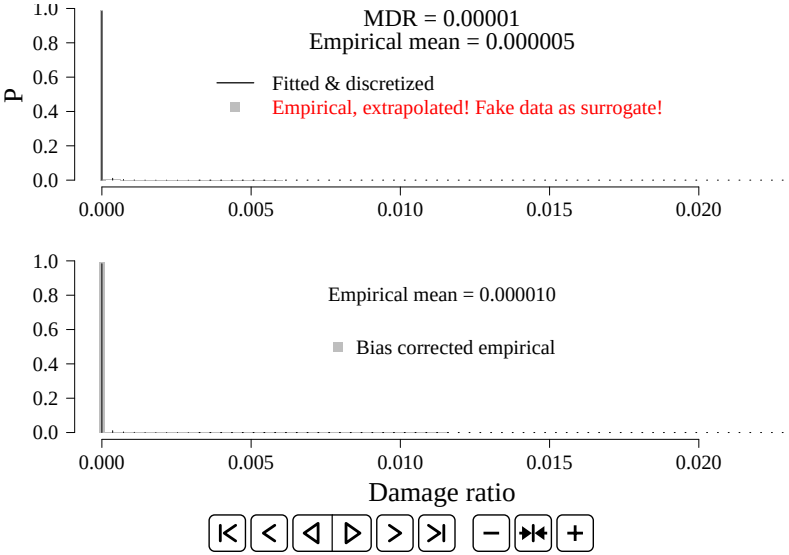


Sliding window size 1784, fitted vs. empirical

Plots are associated with 45 MDRs $\in$ { 1e-5, 2e-5, . . ., 1e-4, 2e-4, . . ., 1e-3, 2e-3, . . ., 1e-2, 2e-2, . . ., 1e-1, 2e-1, . . ., 9e-1 }

"Bias corrected empirical" refers to empirical PMF with support scaled to match mean and MDR.

Bias corrected empirical **is not** used for training TrBs.



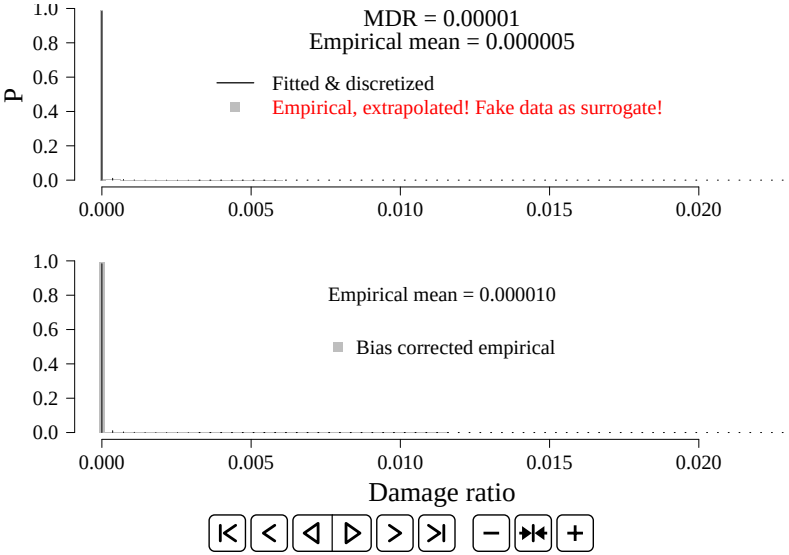
Sliding window size 1784, fitted vs. empirical

Plots are associated with 45 MDRs $\in$ { 1e-5, 2e-5, . . ., 1e-4, 2e-4, . . ., 1e-3, 2e-3, . . ., 1e-2, 2e-2, . . ., 1e-1, 2e-1, . . ., 9e-1 }

"Bias corrected empirical" refers to empirical PMF with support scaled to match mean and MDR.

Bias corrected empirical **is not** used for training TrBs.

Some empirical's mean differs from MDR by orders of magnitude.



Sliding window size 1784, fitted vs. empirical

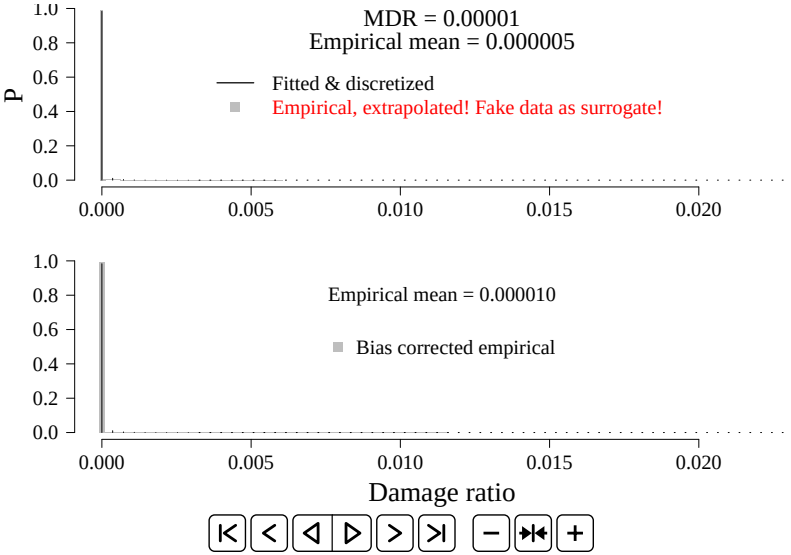
Plots are associated with 45 MDRs $\in$ { 1e-5, 2e-5, . . ., 1e-4, 2e-4, . . ., 1e-3, 2e-3, . . ., 1e-2, 2e-2, . . ., 1e-1, 2e-1, . . ., 9e-1 }

"Bias corrected empirical" refers to empirical PMF with support scaled to match mean and MDR.

Bias corrected empirical **is not** used for training TrBs.

Some empirical's mean differs from MDR by orders of magnitude.

The final discretized TrB often well fits the bias corrected empirical. This suggests d is the dominant parameter for fitting empirical PMF with mean highly different from MDR.



Sliding window size 200, fitted vs. empirical

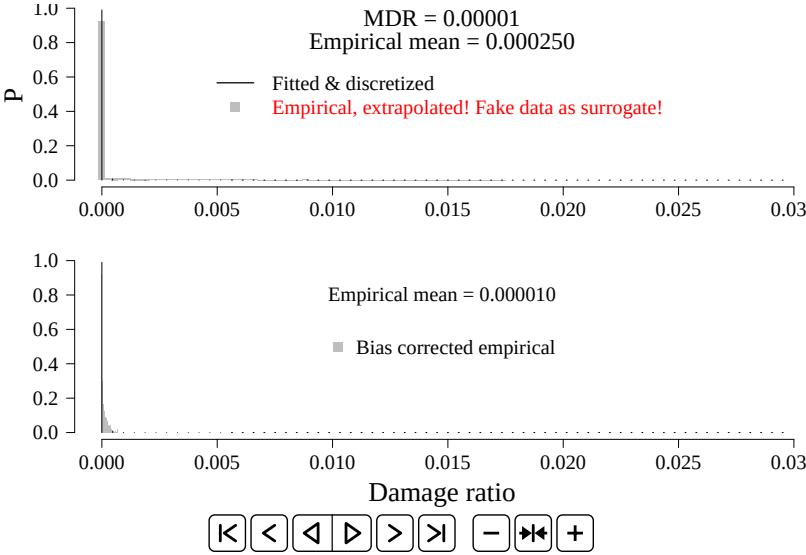
Plots are associated with 45 MDRs $\in$ { 1e-5, 2e-5, . . ., 1e-4, 2e-4, . . ., 1e-3, 2e-3, . . ., 1e-2, 2e-2, . . ., 1e-1, 2e-1, . . ., 9e-1 }

"Bias corrected empirical" refers to empirical PMF with support scaled to match mean and MDR.

Bias corrected empirical **is not** used for training TrBs.

Comparing to window size 1784,

- Shape of empirical PMF is much jaggier due to smaller sample size. Bias between mean and MDR are often larger.
- The final discretized TrB less often well fits the bias corrected empirical.

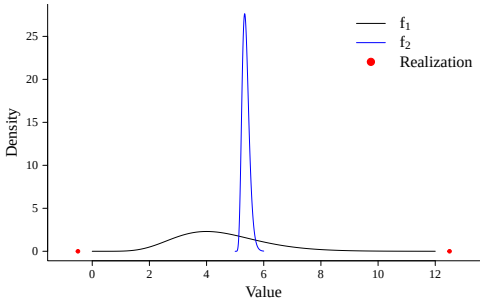


Brier score

The Brier score is appropriate for binary and categorical outcomes that can be structured as true or false, but it is inappropriate for ordinal variables which can take on three or more values.

Have not deciphered why Wikipedia made the above comment.

Without extrapolation, there could be bias towards realizations out of support:

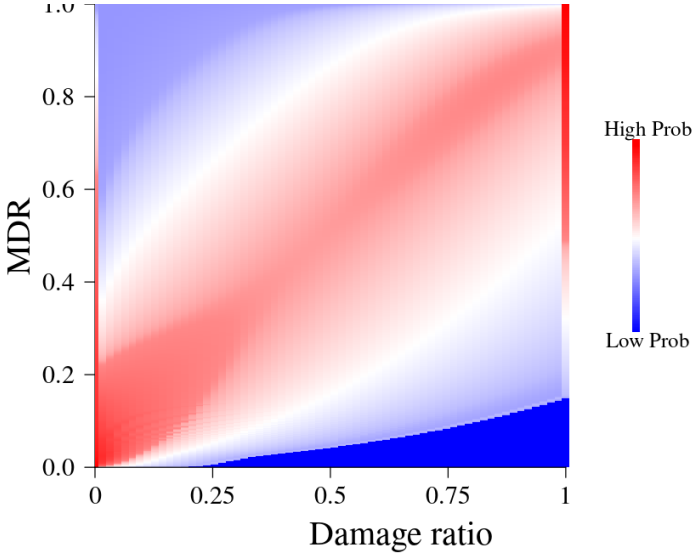


f_1 and f_2 have the same score 0. Fair to say f_2 and f_1 are equally good forecasts?

f_1 has (much) higher chance of being better.

Negative log-likelihood and ignorance score

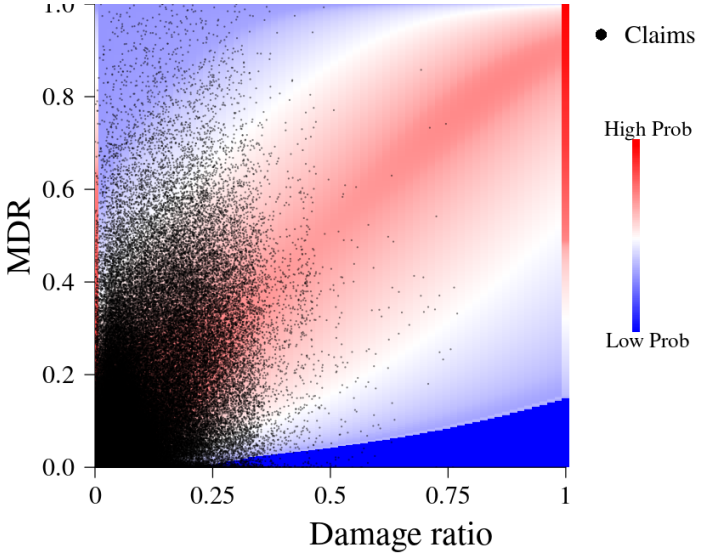
Using claims data, scoring a PMF table $\equiv$ Scoring a 2D joint PMF.



Negative log-likelihood and ignorance score

Using claims data, scoring a PMF table $\equiv$ Scoring a 2D joint PMF.

Claims' MDRs are round to the nearest 1e-5 if < 0.1, and to 1e-4 otherwise.



Negative log-likelihood and ignorance score

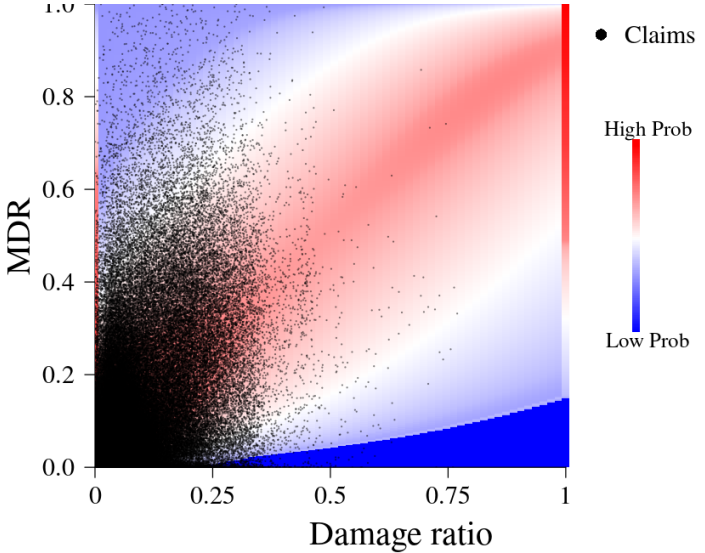
Using claims data, scoring a PMF table $\equiv$ Scoring a 2D joint PMF.

Claims' MDRs are round to the nearest 1e-5 if < 0.1, and to 1e-4 otherwise.

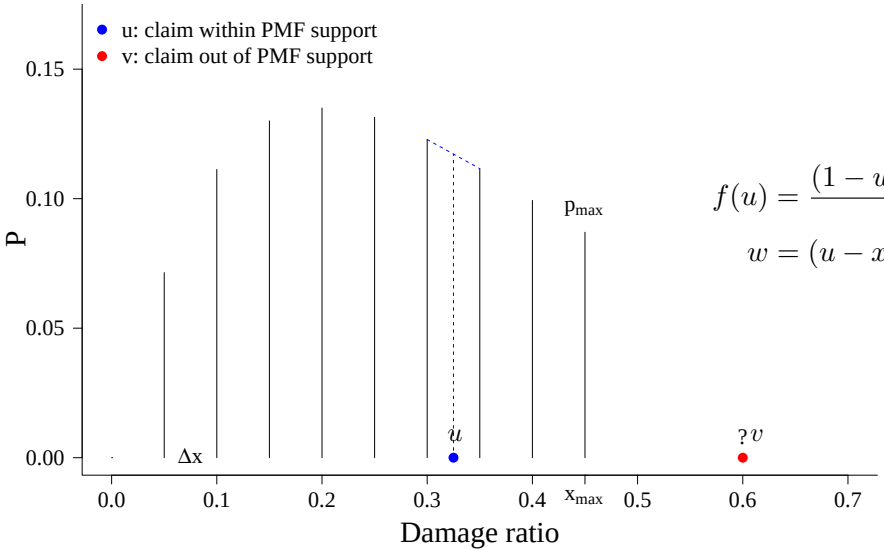
Negative log-likelihood
$\mathcal{L} = -\sum_{i=1}^{N} \ln f(\text{claim}_i) + N \ln 18999.$

Ignorance score $\mathcal{I}g =$
$-\sum_{i=1}^{N} \log_2 f(\text{claim}_i) + N \log_2 18999.$

$N \ln 18999$ and $N \log_2 18999$ account for probability normalization.



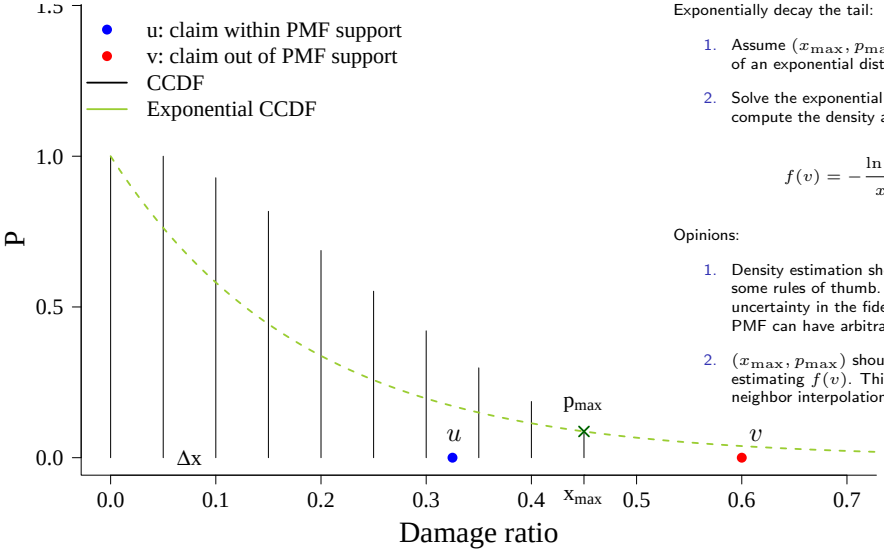
Density estimate of one realization given arbitrary PMF



$$f(u) = \frac{(1-w)\cdot p(x_i) + w \cdot p(x_{i+1})}{\Delta x}$$

$$w = (u - x_i)/(x_{i+1} - x_i)$$

Density estimate of one realization given arbitrary PMF



Exponentially decay the tail:

1. Assume $(x_{\max}, p_{\max})$ is a point on the survival curve of an exponential distribution.
2. Solve the exponential distribution's parameter and compute the density at v:

$$f(v) = -\frac{\ln p_{\max}}{x_{\max}} \exp\left\{\frac{\ln p_{\max}}{x_{\max}} \cdot v\right\}$$

Opinions:

1. Density estimation should be simple and only based on some rules of thumb. More sophistication means more uncertainty in the fidelity of the estimate considering the PMF can have arbitrary shape.
2. $(x_{\max}, p_{\max})$ should be the dominant parameter for estimating $f(v)$. This follows the principle of nearest neighbor interpolation used for computing $f(u)$.