CITY UNIVERSITY LONDON

CSR Building confidence in a computerised world
www.csr.city.ac.uk

Dealing with epistemic uncertainty

in probabilistic assessment of systems
for which high confidence in very high dependability is required

two intriguing results

examples of current work at the
Centre for Software Reliability, City University London

Lorenzo Strigini

IFIP WG 10.4 research report, Sorrento 27 Jan 2014

Background about the Centre for Software Reliability at City

- Founded in 1983 to deal with problems surrounding the [un]reliability of software
  - quickly expanded into a wider "systems" viewpoint, dependability (including security) of socio-technical systems
- about 15 members
- distinctive features
  - emphasis on rigorous assessment (esp. probabilistic)
    - + developing models for empirical assessment as well as for insight
  - dealing with complexity of evidence
    - + exploration of *assurance cases* and ways to make them more rigorous
  - interdisciplinary approach with social sciences
  - extensive work on redundancy and diversity
  - work with industry and regulators, e.g. relationship with nuclear safety research; collaborations with Adelard, a safety consultancy

Recent or ongoing projects: examples

- EU: SESAMO (2012-2015) (Security and Safety Modelling): integrating security and safety assessment in embedded systems, integrating into model driven development
- EU: AFTER (2012-2014) (A Framework for electrical power systems vulnerability identification, defence and restoration)
- U.K. DISPO (for the Control and Instrumentation Nuclear Industry Forum): assessment of software based, diverse protection systems
- U.K. UnCoDe (Uncertainty and Confidence in regulatory Decision making)
- PIA:FARA (2009 - 2010) (Probabilistic Interdependency Analysis: framework, data analysis and on-line risk assessment)
- UK: security analysis of ERTMS specification
- UK: DSTL - challenges of the next 25 years
- UK: Cancer Research UK: assessing computer aided cancer detection
- UK: INDEED (2006-2010) (Interdisciplinary Design and Evaluation of Dependability)
- EU: AMBER (2008-2009) (Assessing, Measuring, and Benchmarking Resilience)
- EU: IRRIIS (2006-2009) (Integrated Risk Reduction of Information-Based Infrastructure Systems)
- EU: ReSIST (2006-2008) (Resilience for Survivability in Information Society Technologies(IST)): roadmapping, E-voting, intrusion tolerance..

L

Technical report: two examples of recent results

at the intersection of two areas of great interest for us

- assessment of highly critical systems - need very high confidence in very low probability of failure
  - e.g. DISPO projects
- how to build argument so as to facilitate the right decisions (authorise operation iff system is safe[/secure] enough)

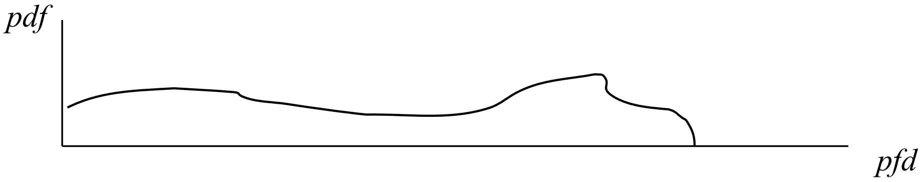
  e.g. UnCoDe project
  - how to describe inevitable uncertainties
  - make decision maker aware of
    - crucial assumptions
    - hidden pitfalls: where in the decision process they should mistrust what seems obvious
  - make things as simple as possible but not simpler

a couple of results:

- when is it that highly precise estimates imply poor dependability?
- worst-case uncertainty and probability of "effectively perfect" software

Background

- applications of interest: want low probability of *any* dangerous failure of subsystem over a duration of operation: e.g., system's lifetime
- we can build probabilistic models that predict probability of any event of interest
  - describing "aleatory" uncertainty: the randomness of the world
- but we have "epistemic" uncertainty. e.g., parameter values are estimated with large uncertainty
  - e.g., probability of failure per demand *(pfd)* of crucial subsystem/ component
- recommended sound method for dealing with this uncertainty (e.g. nuclear PRA):
  - describe the uncertainty as a probability distribution



- in practice, drastic simplifications may be applied
  - use the *expected* value of the distribution
  - guess and force into mathematically tractable distributions

Estimating *pfd* distribution, and avoiding overconfidence

Standard advice

you may have a good hunch / evidence about the true *pfd*

- e.g. a count of previous failures over many systems and much operational experience
- difficult to tell the *spread* of *pfd* among these
- so, *don't be* overconfident (most people are!)
  - don't state too narrow a distribution

avoiding overconfidence.. the surprising result

For the probability of having no failures/accidents, broader distributions
(in a mathematically precise sense of “broader”)
give *higher* probability (optimistic)
(for a broad class of reliability functions)

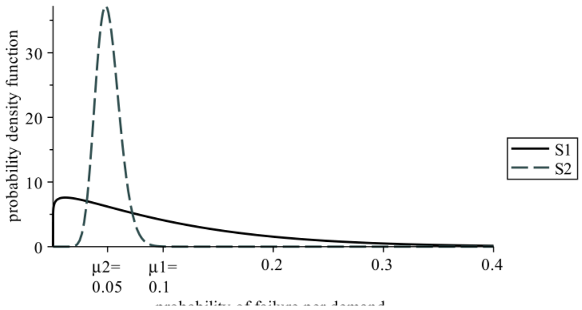
So,

- the “naive”, frequent simplification of using the mean.. is conservative!
  - and other convenient, tighter bounds are available - see paper
- in certain circumstances, a system with less predictable *pfd* gives *lower* risk
  - even if the alternative has better (lower) mean *pfd*

  - this runs against instinct of most engineers and decision makers
  - may create unexpected decision dilemmas in some concrete situations
  - much advisory material about PRA/PSA needs a safety warning

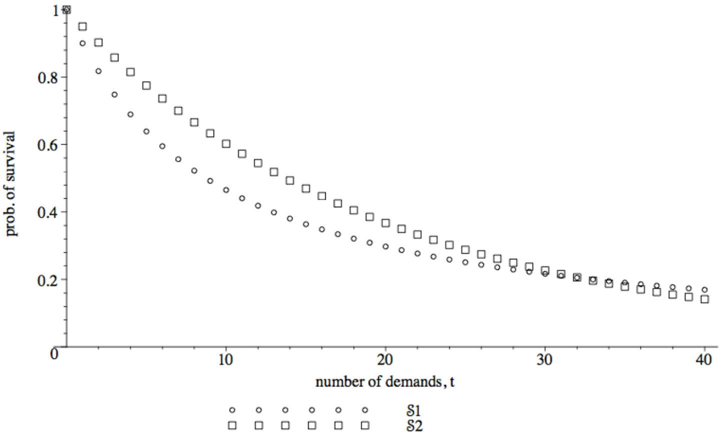
Lower mean *pfd vs* broader distribution

e.g. with these distributions (probability density functions) of the pfd for hypothetical systems S1 and S2, S1 has mean pfd µ1 = 0.1; S2 has lower mean pfd, µ2 = 0.05



... the true probabilities of surviving *t* demands are these:

twice-as-bad mean *pfd* with a wide distribution “wins” in the long run!



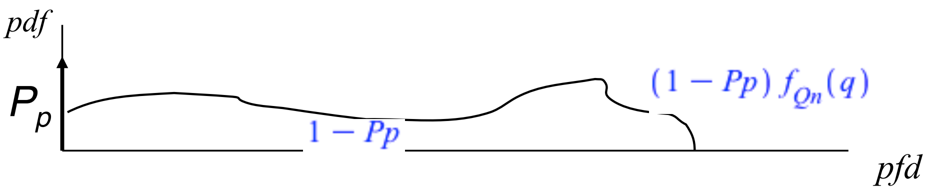
2nd intriguing type of results: worst case inference given epistemic uncertainty

- again, software with requirement of low probability of certain failures over whole system lifetime
- suppose we have *some* probability that the software is good enough
- e.g. *pfd* ≤10^{-9} for top-criticality civil avionics functions

  (where is the evidence?) *Most of the evidence actually supplied !*

  It is about a probability of software having ***no critical faults***

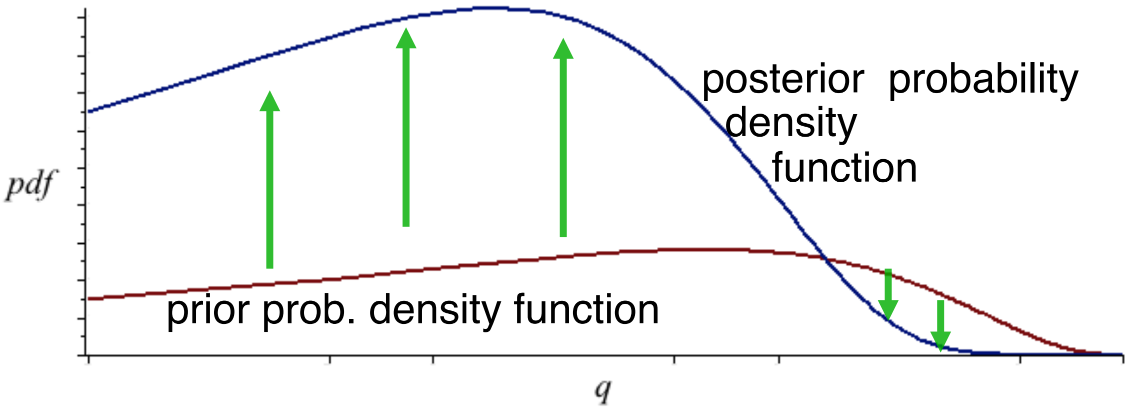
--> given ***will*** to collect statistics, reasonable *Pp* claims can be made



- and to that you add operational experience (testing and real use) and perform Bayesian updating to improve your confidence
- the real difficulty is *the rest* of the distribution

Bayesian inference, in brief

- from *prior* distribution of the random variable of interest: the *pfd* (called *q* for brevity below*)*
- given more evidence, e.g. failure-free processing of demands
- the prior distribution is scaled according to the likelihoods of observing that evidence, conditional on each value of the variable

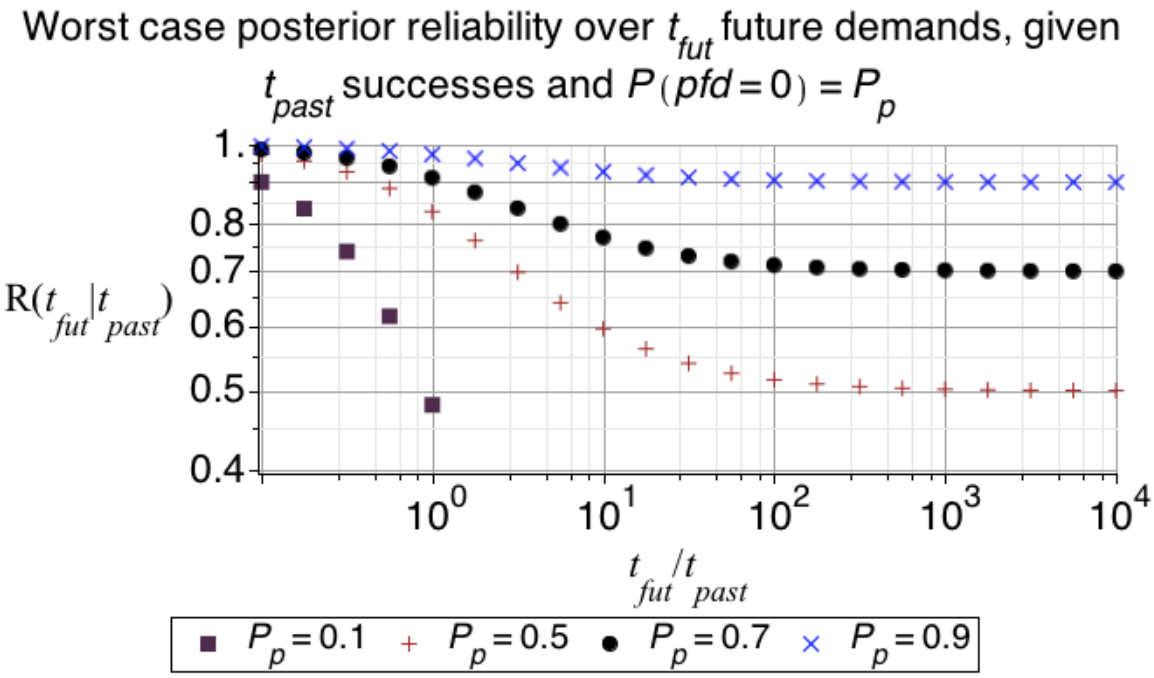


the result: there exists a *worst-case* distribution

.. that given a certain P_p and t_{past} operational successes ensures the most pessimistic possible prediction of probability of surviving t_{fut} future demands in the same environment

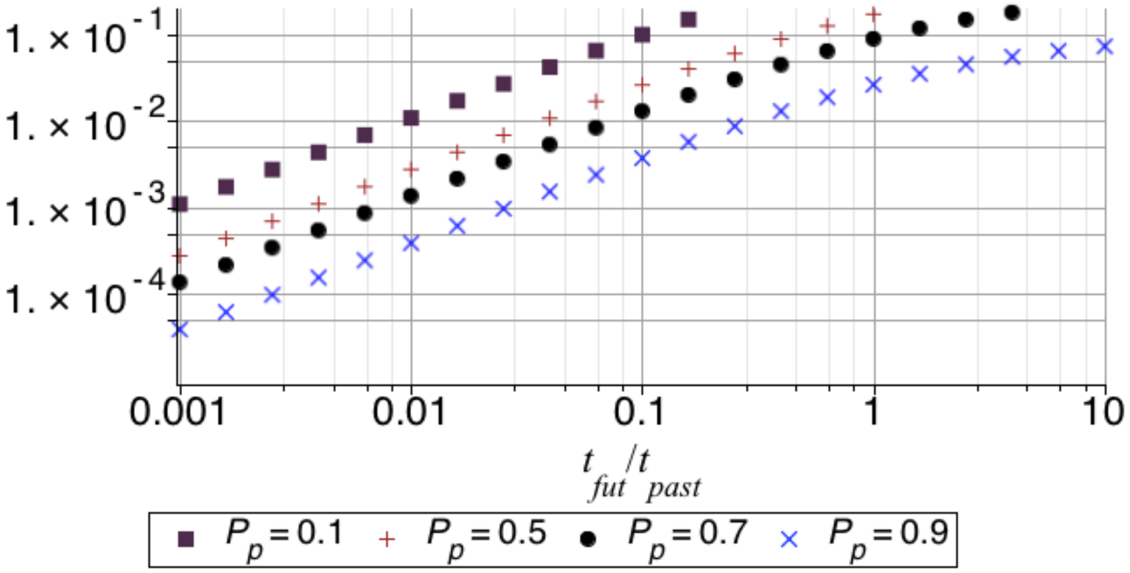
- a bounding result helps to clarify a problem
- and in some concrete scenarios this bound is useful in practice: *not too pessimistic*

Results: worst case posterior reliability



Worst case posterior 1-R(t_{fut}): probability of failing at least once

same information as reliability, but magnifying the region of interest:
reliability close to 1



Thank you for your attention!

Any comments, questions?

For details see:

Strigini, Wright, ”Bounds on survival probability given mean probability of failure per demand; and the paradoxical advantages of uncertainty, 2013, http://openaccess.city.ac.uk/1644/

Povyakalo, Strigini, “Software fault-freeness and reliability predictions”, 2013, http://openaccess.city.ac.uk/2457/

and references therein

more on related work at http://www.csr.city.ac.uk , full text at http://openaccess.city.ac.uk/cgi/search/advanced?screen=Public&PrintSearch&divisions=IICSWR&_action_search=Search