Accurate Approximate Computation --- an example

Derek Partridge & the critical systems group,

School of Engineering, Computer Science and Mathematics

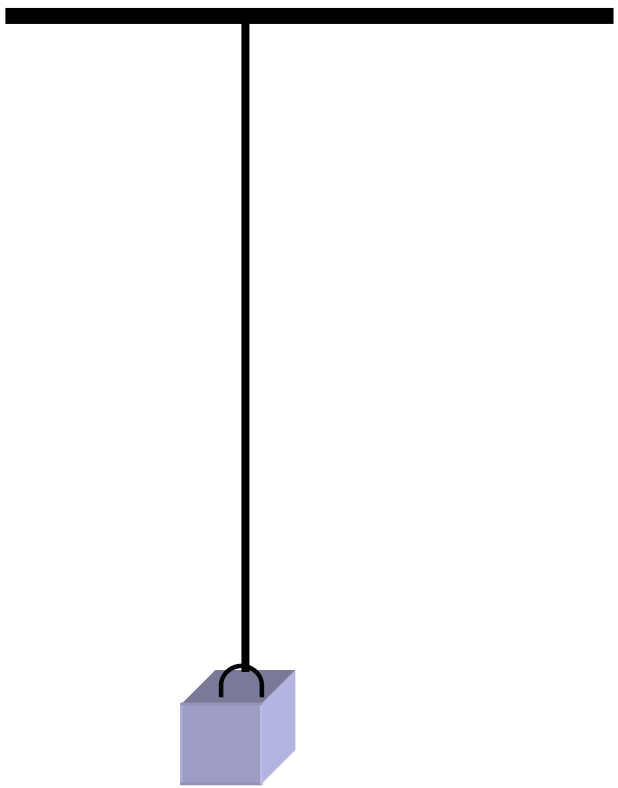
motivations behind AAC

- binary *correct*/*incorrect* basis for computation→efficiency but brittleness
- logical basis promises possibility of provable correctness→practicalities seriously undermine this promise
- undesirable alternative to *correct* can be difficult to identify

in pictures

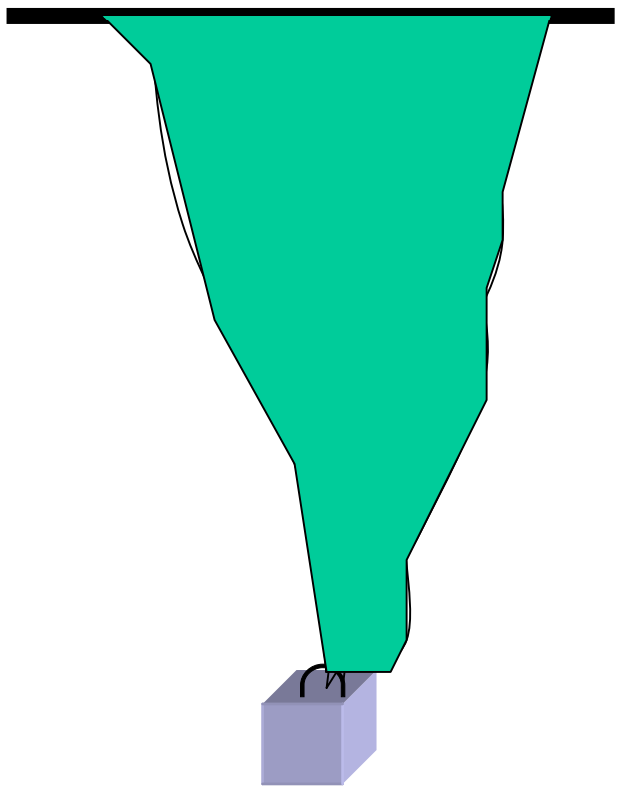
(thus saving thousands of words)

elegant, but fragile
precision



`messy', but robust

accurate approximation



UNIVERSITY of EXETER

a science of ACCURATE APPROXIMATION COMPUTATION

basic quantities not discrete values but probability densities

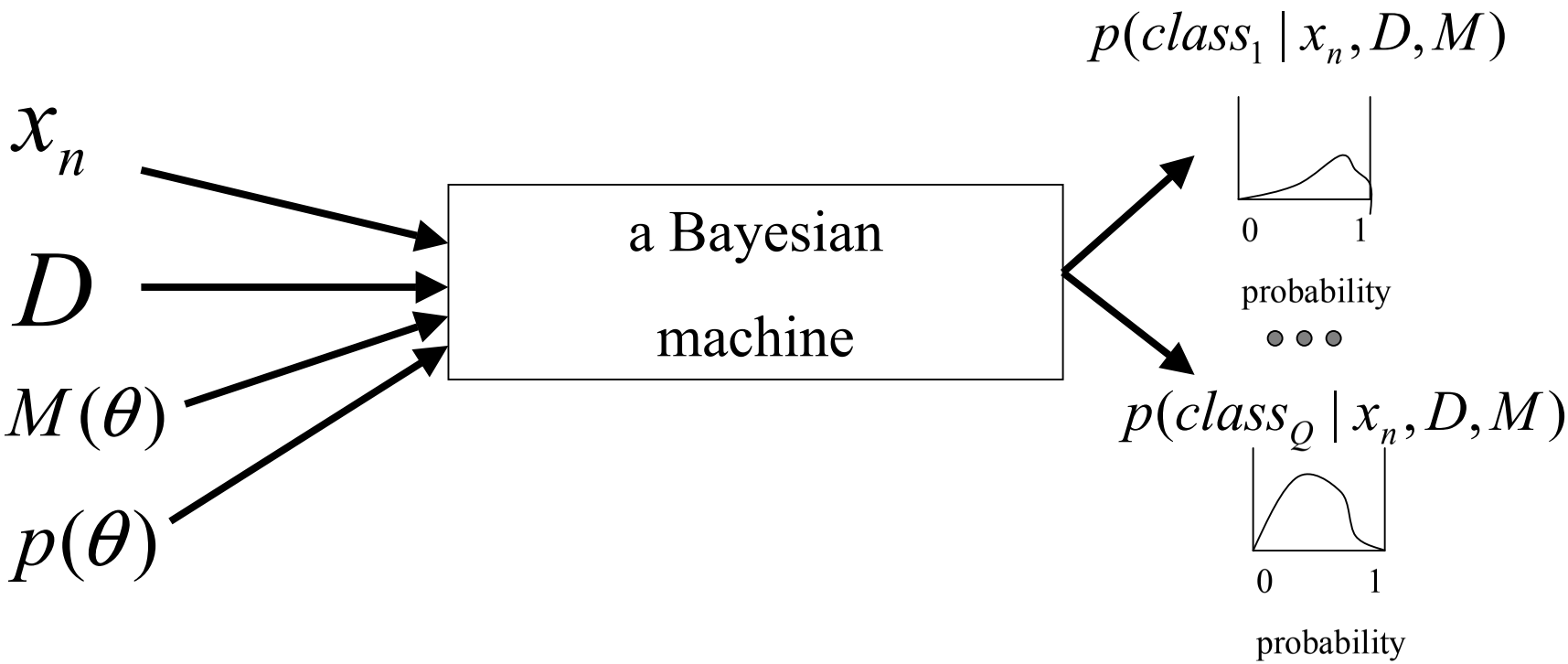
coherent strategy for combining probabilistic components (e.g. Bayes)

results not *correct/incorrect* but accurately approximate

accuracy from massive sampling not verification

statistics rather than formal logic

an example:



the classification of a feature vector x_n

Q possible classes

Simulation on a classical Turing machine

$$p(\theta \mid D, M) = \frac{p(D \mid \theta, M)\, p(\theta \mid M)}{p(D \mid M)}$$

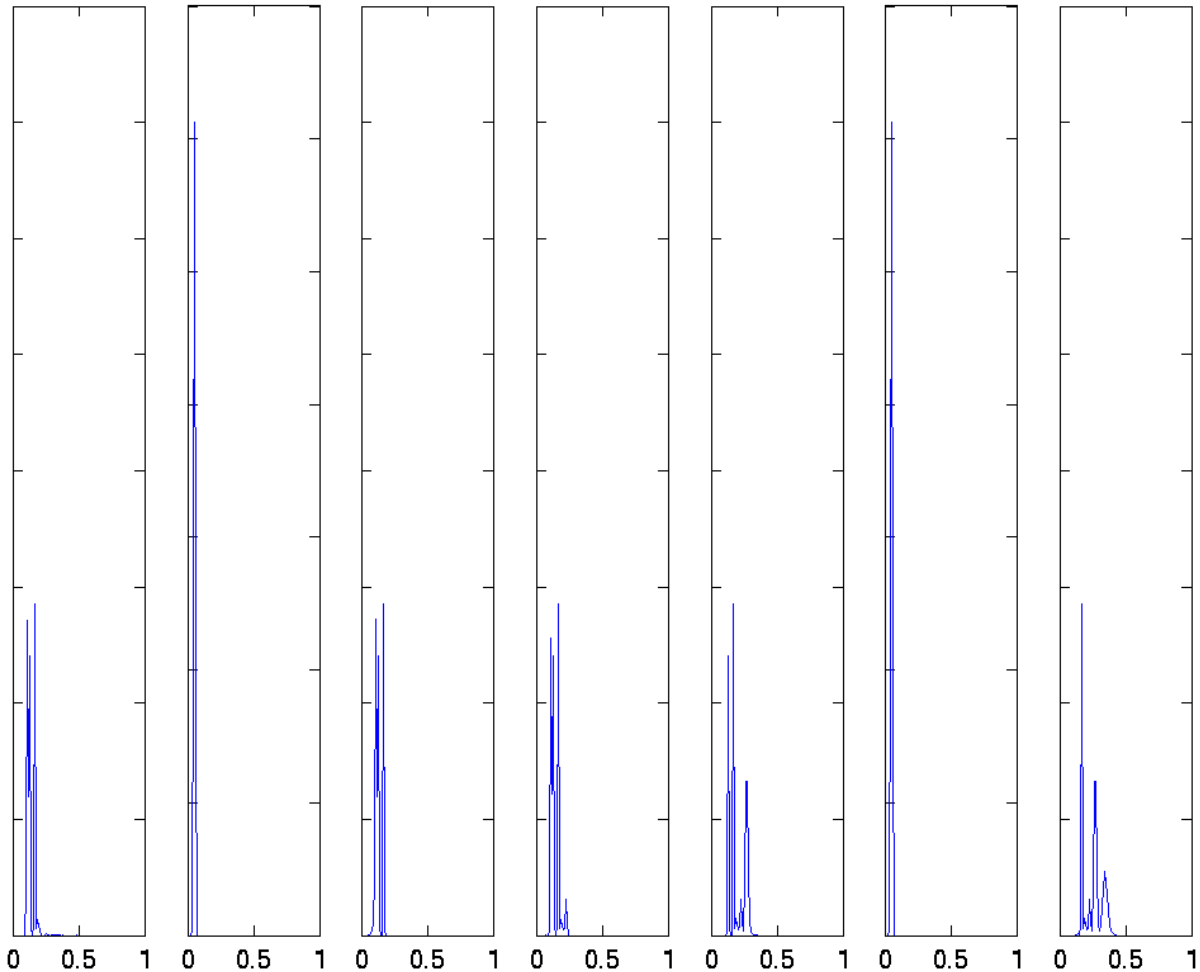
$$p(y \mid x, D, M) = \int p(y \mid x, \theta, \mathrm{M})\mathrm{p}(\theta \mid \mathrm{D}, \mathrm{M})\mathrm{d}\theta$$

using MCMC methods we can draw samples θ^i from $p(\theta \mid D, M)$

$$p(y \mid x, D, M) \approx \frac{1}{N} \sum_{i=1}^{N} p(y \mid x, \theta^{(i)}, M)$$

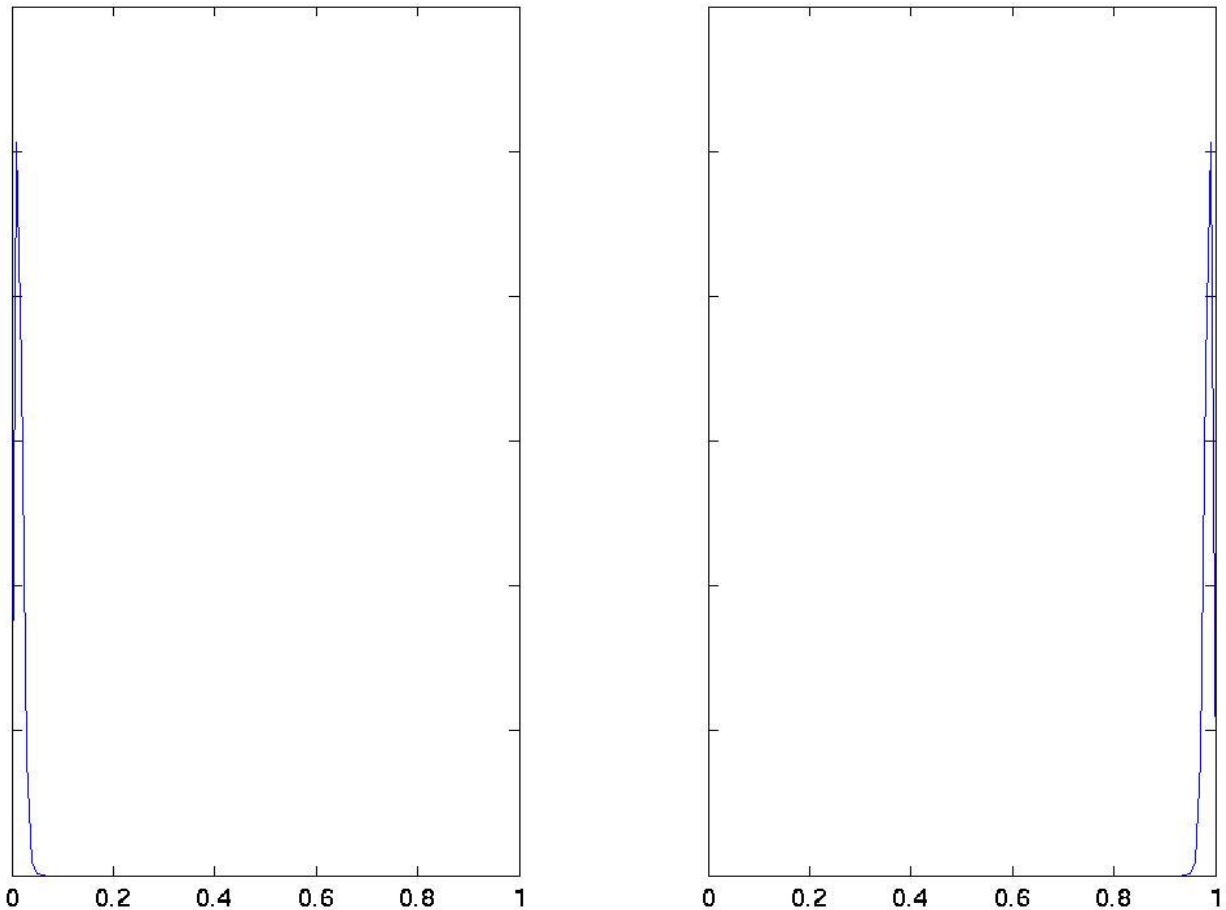
T.C.Bailey, R.M. Everson, W.J.Krzanowski, A. Hernandez, J. Fieldsend and V. Schetinin

UCI ML database: Image data



estimates for each of the 7 output classes for one input vector

UCI ML database: Wisconsin data



estimates of the 2 target classes for 1 input vector:

88.6% SURE correct & 11.4% UNSURE vs 98.3% correct

In sum:

A Bayesian computer is ONE concrete example of Accurate Approximate Computation, and benefits of a 'soft' continuous classification are obvious

But there is a cost in the computational effort of integration, or is there?

Grand Challenge?

Create a science of AAC in the abstract;

Find/Construct an AAC `machine'

meanwhile we work on extending and improving efficiency

of our simulations of AAC examples in classical environment