Entropy Estimation for Non-IID Sources

Kerry McKay
kerry.mckay@nist.gov

Random Bit Generation Workshop 2016

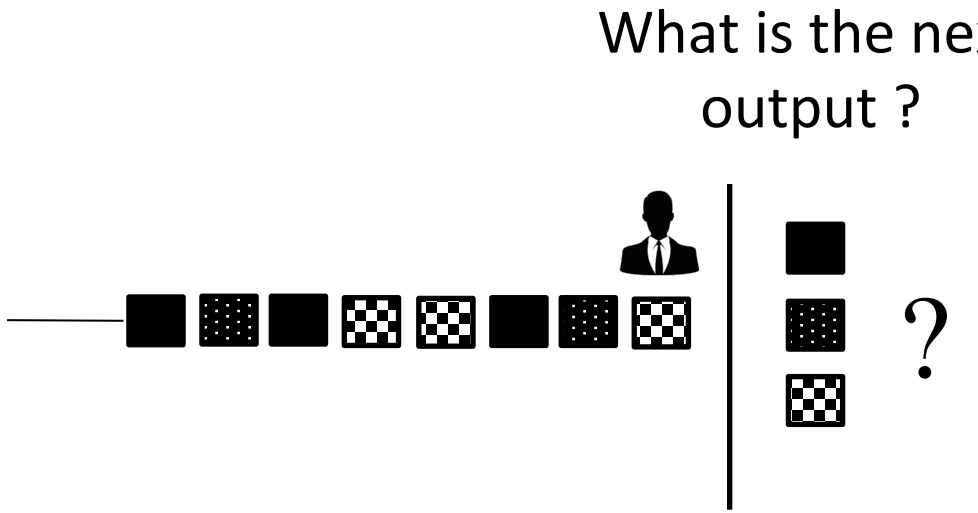
2012 Recap

- 2012 draft of SP 800-90B included non-IID estimators based on entropic statistics
  - Theoretical bounds on IID data
- The methods (tests) were
  - Collision
  - Partial collection (removed)
  - Compression (altered s.d. calculation)
  - Markov
  - Frequency (removed, use Most Common Value estimate instead)
- For all, changed from 95% to 99% confidence interval in 2016

Why Add More?

- There were gaps in 2012 methods
- We wanted to add estimators that were designed for IID and non-IID data that wouldn't unfairly lower entropy estimates
  - Partial collection was often cruel to non-binary sources
- Two types added in 2016 draft
  - Predictors
  - Tuple-based estimates

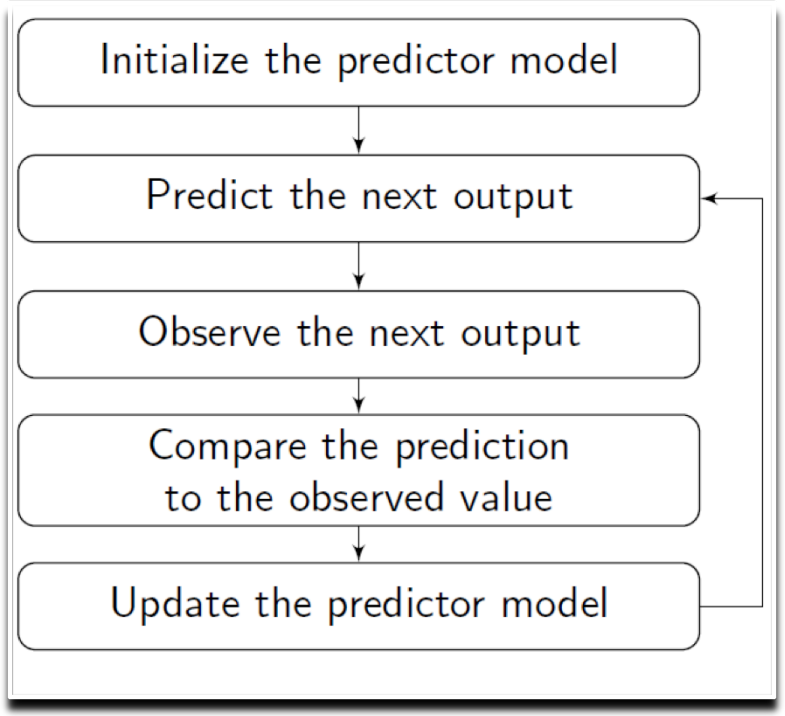
Predictability and Entropy



- Shannon first investigated the relationship between entropy and predictability in 1951
- Used the ability of humans to predict the next character in the text to estimate the entropy per character

Predictors

- Predictors are a framework
- Attempt to mimic adversary that has access to outputs only
- Predictor = model + prediction function
- Given past observations, try to guess next output
- If guess is correct, record 1; else, record 0
- Include last observation in the model



Benefits

- No need to violate assumptions about source's underlying probability distribution
- Can account for changes over time
- Multiple ways of estimating entropy

Estimating Entropy

- After *N* predictions, have a sequence of 1's and 0's
- Interpret sequence as result of *N* independent Bernoulli trials
- We use two notions of predictability to derive entropy estimate from sequence
  - Global predictability
  - Local predictability

Global Predictability

- Considers how well a predictor is able to guess next output on average
- P_{global} = (# correct predictions)/N
- P'_{global} is upper bound of 99% confidence interval on P_{global}

- Pretty straightforward

Local Predictability

- Considers how well a predictor is able to guess next output based on longest run of correct predictions
- Useful if the entropy source falls into highly predictable state
  - What if the DRBG were seeded from a predictable stream of outputs?

- We want to find probability of success for each trial, P_{local} , that is consistent with our observations
- Specifically, we want to find P_{local} such that the probability that we observed the longest run of successes in N trials is 0.99

Local Predictability (cont.)

- Have an asymptotic approximation that tells us the probability that there are no runs of length *r* in *N* trials, given P_{local}
- We turn this around by performing binary search on P_{local} until result is sufficiently close to 0.99
  - Let *r* be length of longest run + 1
  - Solve for P_{local}

$$0.99 = \frac{1 - P_{local}x}{(r + 1 - rx)q} \times \frac{1}{x^{N+1}}$$

  - Where
    - *q* is 1-P_{local}
    - *x* is root of polynomial that can be approximated by iterating a recurrence relation

Ref: Feller, W.: An Introduction to Probability Theory and its Applications, vol. 1, chap. 13. John Wiley and Sons, Inc. (1950)

Predictor Min-Entropy Estimate

- The min-entropy estimate for a predictor is

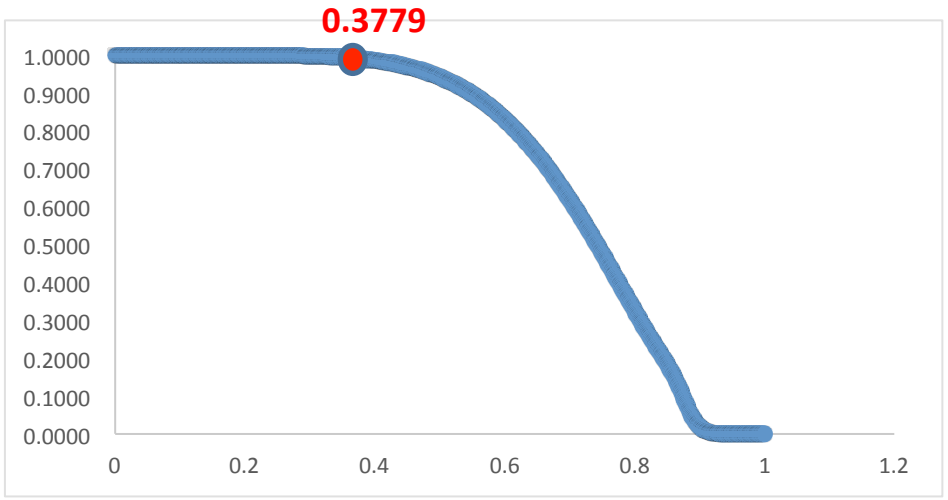
  $-\log_2(\max(P'_{global}, P_{local}))$

- We expect most min-entropy estimates to be based on global predictability
  - Local predictability is intended for severe failures

Example

- Suppose that 14 of 20 guesses were correct
  - $P_{global} = 0.7$
  - $P'_{global} = 0.7+2.576*\text{sqrt}(0.7*0.3/19) = 0.9708$
- Suppose that the longest run of correct guesses is 6
  - Binary search finds that $P_{local} = 0.3779$
- $P'_{global} > P_{local}$
- Min-entropy estimate is

$$-\log_2(P'_{global}) \approx 0.0428$$



Ensemble Predictors

- Several predictors can be combined into one
  - E.g., different parameters for model construction and/or prediction function
  - Call each one a subpredictor
- Ensemble predictor keeps track of performances of each subpredictor in a scoreboard
- Best performing subpredictor is used for the next prediction
- The final entropy estimate is based on success of the ensemble predictor, not on the individual performance of the subpredictors

90B Predictors

- In SP 800-90B strategy (take lowest estimate), a predictor will only lower the awarded entropy estimate if it is good at guessing the next output
  - Bad models can't significantly lower the estimate
- Without source knowledge, difficult to make best predictor
  - We can make generic predictors that perform reasonably well

90B Predictors

- SP 800-90B specifies four generic predictors:
  - Multi Most Common in Window Prediction
  - Lag Prediction
  - MultiMMC Prediction
  - LZ78Y Prediction

- MultiMCW, Lag, and MultiMMC are ensemble predictors

Multi Most Common in Window Predictor

- Each subpredictor keeps window of previous *w* observations
  - We use four window sizes *w*=63, 255, 1023, and 4095
  - Prediction is the most common value in the window
- Performs well in cases where there is a clear most common value, but the value may vary over time
  - E.g., due to environmental conditions such as operating temperature

Lag Predictor

- Each subpredictor predicts value observed at a fixed lag, *d*
  - Example: if *d*=1, the subpredictor predicts the last observed value
- 90B lag predictor contains 128 subpredictors for lags from 1 to 128
- Performs well on sources with strong periodic behavior, if *d* is related to period

MultiMMC Predictor

- Multiple Markov Model with Counting
- Each subpredictor constructs a Markov model from observed outputs
  - Records the observed frequencies of transitions (rather than probabilities)
  - Prediction follows most frequently observed transition from the previous *d* outputs
- MultiMMC ensemble predictor uses 16 Markov models with order from 1 to 16
- Works well on sources where outputs are dependent on previous 16 or fewer outputs

LZ78Y Predictor

- Shares concepts with MultiMMC, but applied differently
  - Both look at previous outputs and build model with counts of next outputs
  - This is not an ensemble predictor
  - Prediction favors longest string with highest count, not length that performed best in the past
  - Model (dictionary) construction is bounded
- Performs well on sources that would be efficiently compressed by LZ78-like compression algorithms

Tuple-based Estimates

- Added two tuple-based estimates that are based on tuples
  - t-tuple estimate
  - LRS estimate
- These tuple estimates attempt to capture global properties of output sequence

t-Tuple Estimate

- Estimate based on frequencies of tuples
- *t* is largest value such that most common *t*-tuple appears at least 35 times in sequence
- For *i* from 1 to *t*, calculate proportion of highest frequency of *i*-tuple to all *i*-tuples in sequence
- P_{max} for each *i* is i^{th} root of proportion
- Entropy is calculated from highest P_{max}

LRS Estimate

- Longest repeated substring
  - Estimates collision entropy
  - LRS concept also appears in IID testing, but does not award entropy estimate
- Find length of smallest repeated substring that occurs < 20 times, *u*
- Find length of longest repeated substring, *v*
- For *W* from *u* to *v*, estimate collision probability and max probability of output
- Use highest max probability to derive min-entropy estimate

Summary

- The non-IID path now includes generic predictors and tuple-based estimates
- Predictors mimic attacker guessing the next output based on previous outputs and simple models
- Tuple-based estimates that capture global properties
- Complement entropic statistics approach