The Luckiness Principle

Peter Grünwald
CWI, Amsterdam

- Ideas from COLT/Information Theory
  - e.g. PAC-Bayes
  - universal data compression

- Model uncertainty using probability distribution. Incorporate prior knowledge, but with different interpretation from Bayesian priors
  - make much weaker claims about the world than a Bayesian predictor does
Example: Gzip

• Consider your favorite data compressor, e.g. Gzip
• There must be some distribution $P_{gzip}$ over byte sequences $x_1, x_2, \ldots$ such that, for all $n, x^n$,
  $$- \log P_{gzip}(x_1, \ldots, x_n) = L_{gzip}(x_1, \ldots, x_n)$$
• Suppose $X_1, X_2, \ldots \sim P^*$
  - If $P^* \neq P_{gzip}$ then there must be a code that is, on average, more efficient than Gzip
  - If $P^* = P_{gzip}$ then Gzip is the most efficient, and the entropy $H(P_{gzip})$ is a good estimate of the average number of bits needed to encode a text, i.e. for real texts, it should hold that
    $$H(P_{gzip}) \approx n^{-1}L_{gzip}(x^n)$$
Example: Gzip

- Consider your favorite data compressor, e.g. Gzip
- There must be some distribution \( P_{gzip} \) over byte sequences \( x_1, x_2, \ldots \) such that, for all \( n, x^n \),
  \[- \log P_{gzip}(x_1, \ldots, x_n) = L_{gzip}(x_1, \ldots, x_n)\]
- Suppose \( X_1, X_2, \ldots \sim P^* \)
  - If \( P^* \neq P_{gzip} \) then there must be a code that is, on average, more efficient than Gzip
  - If \( P^* = P_{gzip} \) then Gzip is the most efficient, and the entropy \( H(P_{gzip}) \) is a good estimate of the average number of bits needed to encode a text, i.e. for real texts, it should hold that
    \[ H(P_{gzip}) \approx n^{-1} L_{gzip}(x^n) \]

But of course, this doesn’t hold at all!
Luckiness Principle

• Gzip (PPMD) distribution perform reasonable if used for some prediction tasks
  – e.g. data compression, estimate frequencies of symbols
• But very bad when used for other tasks
  – e.g. estimate how many bits you need if you use it to code data
• Apparently, prior knowledge is “adequate” for some prediction tasks, not others
  → different construction of “predictive distributions”:

\[
\arg\min_{\tilde{P}} \max_{x^n \in \mathcal{X}^n} \left\{ -\log \tilde{P}(x^n) - \left[ \min_{\theta \in \Theta} -\log P(x^n | \theta) + a(\theta) \right] \right\}
\]

\[
\tilde{P}(x^n) \approx \sum_{\theta \in \Theta} P(x^n | \theta) \frac{e^{-a(\theta)}}{\sum_{\theta' \in \Theta} e^{-a(\theta')}}
\]
Question

- I indicated how to do this for log loss/data compression (modern versions of MDL are actually based on this idea!)
- But can we come up with a formulation for general loss functions?
- ...apologies for not being practical...