Offline to Online Conversion

Marcus Hutter

Canberra, ACT, 0200, Australia
http://www.hutter1.net/

ANU
THE AUSTRALIAN NATIONAL UNIVERSITY

Abstract

I consider the problem of converting offline estimators into an online predictor or estimator with small extra regret. Formally this is the problem of merging a collection of probability measures over strings of length 1,2,3,... into a single probability measure over infinite sequences. I describe various approaches and their pros and cons on various examples. As a side-result I give an elementary non-heuristic purely combinatoric derivation of Turing's famous estimator. My main technical contribution is to determine the computational complexity of online estimators with good guarantees in general.

Keywords: offline, online, batch, sequential, probability, estimation, prediction, time-consistency, tractable, regret, combinatorics, Bayes, Laplace, Ristad, Good-Turing.

Contents

- Introduce and discuss the problem of converting offline estimators (q_n) to an online predictor $\tilde{q}$.
- Define the worst-case extra regret R_n of online $\tilde{q}$ over offline (q_n) estimator, measuring the conversion quality.
- Compare and discuss the pros and cons of four offline-to-online conversion proposals (rat,n1,lim,mix).
- Illustrate their behavior for various classical estimators (Bayes, MDL, Laplace, Good-Turing, Ristad)
- Computational complexity of online estimators with good guarantee.
- Open problems regarding efficient low-regret online estimators.
- Simple and non-heuristic derivation of the famous Good-Turing estimator.

PROBLEM FORMULATION

Problem Formulation (Online=TC=Norm)

Notation: $x_{t:n} := x_t ... x_n \in \mathcal{X}^{n-t+1}$, $\quad x_{<n} := x_1 ... x_{n-1}$, $\quad x_{1:0} = x_{<1} = \epsilon$.

Formulation 1 (measures)

- Given: Probability measures Q_n on $\mathcal{X}^n$ for $n = 1, 2, 3, ...$.
- Seeked: Online probability measure $\tilde{Q}$ on $\mathcal{X}^\infty$ close to all Q_n in the sense of $\tilde{Q}(\mathcal{A} \times \mathcal{X}^\infty) \approx Q_n(\mathcal{A})$ for all measurable $\mathcal{A} \subseteq \mathcal{X}^n$ and all n.

Formulation 2 (probability mass function for finite $\mathcal{X}$)

- Given: Prob. mass functions $q_n : \mathcal{X}^n \to [0;1]$, i.e. $\sum_{x_{1:n}} q_n(x_{1:n}) = 1$.
- Seeked: Time-consistent (TC) fct. $\tilde{q} : \mathcal{X}^* \to [0;1]$ with
  $\sum_{x_n} \tilde{q}(x_{1:n}) = \tilde{q}(x_{<n})\ \forall n, x_{<n}$ and $\tilde{q}(\epsilon) = 1$
  close to q_n i.e. $\tilde{q}(x_{1:n}) \approx q_n(x_{1:n})$ for all n and $x_{1:n}$.

Formulation 3 (predictors)

- Seeked: Normalized (Norm) predictor $\tilde{q} : \mathcal{X} \times \mathcal{X}^* \to [0;1]$ with
  $\sum_{x_n} \tilde{q}(x_n|x_{<n}) = 1 \quad \forall n, x_{<n}$ such that its joint probability
  $\tilde{q}(x_{1:n}) \ := \prod_{t=1}^n \tilde{q}(x_t|x_{<t})$ is close to q_n as before.

Discussion: $\tilde{q}(x_{1:n})$ is prob. that an (infinite) sequence starts with $x_{1:n}$.
$\tilde{q}(x_n|x_{<n}) \equiv \tilde{q}(x_{1:n})/\tilde{q}(x_{<n})$ is the probability that x_n follows given $x_{<n}$.

Example Applications

(i) To use an offline estimator (q_n) to make stochastic predictions we need to expand and normalize it.

(ii) Maximum likelihood estimation $\hat{\theta}_n$ of parameter $\theta \in \Theta$ leads to offline estimator $(q_n) := (q^{\hat{\theta}_n})$ even if q^θ was online for all θ.

(iii) Arithmetic coding requires an online estimator,
but is often based on a class of distributions as described in (ii).

(iv) Computing the cumulative distribution function $\sum_{y_{1:n} \leq x_{1:n}} q_n(y_{1:n})$
can be hard for an offline estimator,
but fast$=O(n)$ if (q_n) is (converted to) online.

Performance/Distance Measure

Natural measure of distance of $\tilde{q}$ from q_n:
Worst-case log-loss regret:

$$R_n \;\equiv\; R_n(\tilde{q}) \;\equiv\; R_n(\tilde{q}||q_n) \;:=\; \max_{x_{1:n}} \ln \frac{q_n(x_{1:n})}{\tilde{q}(x_{1:n})}$$

Properties:

- Quantifies $\tilde{q} \approx q_n$.
- $R_n \geq 0$, and $R_n = 0$ iff $\tilde{q}_{|\mathcal{X}^n} = q_n$.
- Online arithmetic code of $x_{1:n}$ w.r.t. $\tilde{q}$ has length $|\log_2 \tilde{q}(x_{1:n})|$.
  Offline Huffman code for $x_{1:n}$ w.r.t. q_n has code length $|\log_2 q_n(x_{1:n})|$.

$$\Longrightarrow \quad \mathrm{CL}^{\text{online}}(x_{1:n}) - \mathrm{CL}^{\text{offline}}(x_{1:n}) \;\leq\; R_n \ln 2$$

Plenty of alternatives: E.g. $\mathrm{KL}(q_n||\tilde{q}) \leq R_n \geq \mathrm{KL}(\tilde{q}||q_n)$ not considered.

Extending q_s from $\mathcal{X}^s$ to $\mathcal{X}^\infty$

It is always possible to choose $\tilde{q} := \bar{q}_s$ such that $R_s = 0$ for some $s \in \mathbb{N}_0$

(but $R_n > 0$ for $n \neq s \implies$ naive minimization of R_n w.r.t. $\tilde{q}$ fails)

$$\bar{q}_s(x_{1:n}) \; := \; \left\{ \begin{array}{cl} q_s(x_{1:s}) & \text{if} \quad n = s, \\ \sum_{x_{n+1:s}} q_s(x_{1:s}) & \text{if} \quad n < s, \\ q_s(x_{1:s}) Q(x_{s+1:n}|x_{1:s}) & \text{if} \quad n > s \end{array} \right.$$

Q can be an arbitrary measure on $\mathcal{X}^\infty$, e.g. uniform $Q(x_{s+1:n}|x_{1:s}) = |\mathcal{X}|^{n-s}$

I now consider four methods of converting offline estimators
to online predictors ...

CONVERSION METHODS

Naive Ratio $\tilde{q}^{\text{rat}}$

The simplest way to define a predictor $\tilde{q}$ from q_n is via Ratio

$$\tilde{q}^{\text{rat}}(x_t|x_{<t}) \;:=\; \frac{q_t(x_{1:t})}{q_{t-1}(x_{<t})} \quad \text{or equivalently} \quad \tilde{q}^{\text{rat}}(x_{1:n}) \;:=\; q_n(x_{1:n})$$

Tractable but obviously only works when q_n already is Online.

Otherwise $\tilde{q}^{\text{rat}}$ violates TC.

Degree of violation = Normalizer:

$$\mathcal{N}(x_{<t}) \;:=\; \sum_{x_t} \tilde{q}^{\text{rat}}(x_t|x_{<t}) \;\equiv\; \frac{\sum_{x_t} q_t(x_{1:t})}{q_{t-1}(x_{<t})}$$

Naive Normalization $\tilde{q}^{\text{n1}}$

Correct failure of $\tilde{q}^{\text{rat}}(x_t|x_{<t})$ to satisfy Norm by Normalization: [Sol78]

$$\tilde{q}^{\text{n1}}(x_t|x_{<t}) \;:=\; \frac{q_t(x_{1:t})}{\sum_{x_t} q_t(x_{1:t})} \;\equiv\; \frac{\tilde{q}^{\text{rat}}(x_t|x_{<t})}{\mathcal{N}(x_{<t})} \quad \text{and}$$

$$\tilde{q}^{\text{n1}}(x_{1:n}) \;:=\; \prod_{t=1}^{n} \tilde{q}^{\text{n1}}(x_t|x_{<t}) \;\equiv\; \frac{q_n(x_{1:n})}{\prod_{t=1}^{n} \mathcal{N}(x_{<t})}$$

For small $\mathcal{X}$ is still tractable but can result in very large regret $R_n \propto n$.

Express and upper bound regret R_n in terms of Normalizer $\mathcal{N}$:

$$R_n(\tilde{q}^{\text{n1}}) \;=\; \max_{x_{1:n}} \sum_{t=1}^{n} \ln \mathcal{N}(x_{<t}) \;\le\; \sum_{t=1}^{n} \ln \max_{x_{<t}} \mathcal{N}(x_{<t})$$

If q_n is TC, then $\mathcal{N} \equiv 1$, hence R_n as well as the upper bound are $\equiv 0$.

Limit $\tilde{q}^{\text{lim}}$

Since $R_s(\bar{q}_s) = 0$ for any fixed s, a natural idea is taking the Limit

$$\tilde{q}^{\text{lim}}(x_{1:n}) \;:=\; \lim_{s\to\infty} \bar{q}_s(x_{1:n}) \;=\; \lim_{s\to\infty} \sum_{x_{n+1:s}} q_s(x_{1:s})$$

in the hope to make $\lim_{s\to\infty} R_s = 0$.

Effectively what $\tilde{q}^{\text{lim}}$ does is to use q_s for very large s also for short strings of length n by marginalization.

Problems are plenty:

- The limit may not exist,
- may exist but be incomputable,
- R_n may be hard to impossible to compute or upper bound,
- and even if the limit exists, $\tilde{q}^{\text{lim}}$ may perform badly.

Mixture $\tilde{q}^{\text{mix}}$

Take Bayesian Mixture over the class $\{\bar{q}_1, \bar{q}_2, ...\}$ of all $\bar{q}_s$ [San06]

$$\tilde{q}^{\text{mix}}(x_{1:n}) \;:=\; \sum_{s=0}^{\infty} \bar{q}_s(x_{1:n}) w_s \quad \text{with prior} \quad w_s > 0, \quad \sum_{s=0}^{\infty} w_s = 1.$$

$\tilde{q}^{\text{mix}}$ is TC and its regret can easily be upper bounded: [San06]

$$R_n(\tilde{q}^{\text{mix}}) \;=\; \max_{x_{1:n}} \ln \frac{q_n(x_{1:n})}{\sum_{s=0}^{\infty} \bar{q}_s(x_{1:n}) w_s} \;\leq \max_{x_{1:n}} \ln \frac{q_n(x_{1:n})}{\bar{q}_n(x_{1:n}) w_n} \;=\; \ln w_n^{-1}$$

For e.g. $w_n := \frac{1}{(n+1)(n+2)}$ we have $\ln w_n^{-1} \leq 2 \ln(n+2)$= small.

Conclusion: **Any** offline estimator can be converted into
an online predictor with very small extra regret.

Problem: How convert this heavy construction into an efficient algorithm?

Variations: $Q \equiv 0$ -or- sparser w_n.

EXAMPLES

Examples by Category

Class of Probabilities (ML/MAP/MDL/NML/MML/Bayes):

- Start with a class $\mathcal{M}$ of probability measures ν on $\mathcal{X}^\infty$ in the hope one of them is good.

Combinatorial (Uniform,Laplace,Good-Turing,Ristad):

- Assigns uniform probabilities over subsets of $\mathcal{X}^n$.

Exponentiated Code Length:

- not further discussed

Bayes

The Bayesian mixture over $\mathcal{M}$ w.r.t. some prior (density) $w()$ is

$$q_n(x_{1:n}) \;:=\; \int_{\mathcal{M}} \nu(x_{1:n})\, w(\nu)\, d\nu$$

q_n is TC $\quad\Longrightarrow\quad (q_n^{\text{rat}}) \equiv (q_n^{\text{n1}}) \equiv (q_n^{\text{lim}}) \equiv \tilde{q} \quad\Longrightarrow\quad R_n = 0.$

$\tilde{q}^{\text{rat}}$ is tractable if the Bayes mixture is.

Assume the true sampling distribution μ is in $\mathcal{M}$:

For countable $\mathcal{M}$ and counting measure $d\nu$, we have
$q_n(x_{1:n}) \;\geq\; \mu(x_{1:n})w(\mu)$, hence $R_n^{\text{online}} = R_n^{\text{offline}} \;\leq\; \ln w(\mu)^{-1}$.

For continuous classes $\mathcal{M}$ under mild conditions:
$R_n^{\text{online}} = R_n^{\text{offline}} \;\lesssim\; \ln w(\mu)^{-1} + O(\ln n)$. [BC91, Hut03, ?, RH07]

ML/MAP/MDL/NML/MML

$$\text{MAP=MDL estimator:} \quad \hat{q}_n(x_{1:n}) \;:=\; \sup_{\nu\in\mathcal{M}} \{\nu(x_{1:n})\, w(\nu)\}$$

$$\text{NML estimator:} \quad q_n(x_{1:n}) \;:=\; \frac{\hat{q}_n(x_{1:n})}{\sum_{x_{1:n}} \hat{q}_n(x_{1:n})}$$

Since $\hat{q}_n$ is not even a probability on $\mathcal{X}^n$,
we have to normalize it to q_n (ML/NML).

Unlike Bayes, q_n is not TC, causing various complications. [Grü07, Hut09]

Crude MDL: $q_n := \arg\max_{\nu\in\mathcal{M}}\{\nu(x_{1:n})\, w(\nu)\}$
is a probability measure on $\mathcal{X}^\infty$ for each n, but also not TC. [PH05]

Uniform

- The uniform probability $q_n(x_{1:n}) := |\mathcal{X}|^{-n}$ is TC

  $\Longrightarrow$ all four $\tilde{q}$ coincide and $R_n = 0\,\forall n$.

- Lousy estimator, since predictor $\tilde{q}(x_t|x_{<t}) = 1/|\mathcal{X}|$ is indifferent and ignores all evidence $x_{<t}$ to the contrary.

- Improvement: Partition the sample space (here $\mathcal{X}^n$) and assign uniform probabilities to and within each partition.

- The Laplace rule can be derived that way, and the Good-Turing and Ristad estimators by further sub-partitioning.

Laplace (Double Uniform)

- $n_i := |\{t : x_t = i\}|$ is number of times, symbol $i \in \mathcal{X} = \{1,...,d\}$ appears in $x_{1:n}$.
- Assign uniform probability to all sequences $x_{1:n}$ with the same counts $\mathbf{n} := (n_1,...,n_d)$, therefore $q_n(x_{1:n}|\mathbf{n}) = \binom{n}{n_1 \ldots n_d}^{-1}$.
- Assign uniform probability to the counts $\mathbf{n}$ themselves, therefore $q_n(\mathbf{n}) = |\{\mathbf{n} : n_1 + ... + n_d = n\}|^{-1} = \binom{n+d-1}{d-1}^{-1}$.
- Together

$$q_n(x_{1:n}) \;=\; \binom{n}{n_1 \;\ldots\; n_d}^{-1}\binom{n+d-1}{d-1}^{-1} = \binom{n+d-1}{n_1 \;\ldots\; n_d \quad d-1}^{-1}$$

$$\Longrightarrow \quad \tilde{q}^{\text{rat}}(x_{n+1} = i|x_{1:n}) \;=\; \frac{q_{n+1}(x_{1:n}i)}{q_n(x_{1:n})} \;=\; \frac{n_i + 1}{n + d}$$

- Is properly normalized (Norm), so $\tilde{q}^{\text{rat}}$ is TC.
- $(q_n^{\text{rat}}) \equiv (q_n^{\text{n1}}) \equiv (q_n^{\text{lim}})$ coincide with $\tilde{q}$ and $R_n = 0$.
- $\tilde{q}^{\text{rat}}$ is nothing but Laplace's famous rule.

Good-Turing (Triple Uniform)

- $M_r := \{i : n_i = r\}$ = symbols that appear exactly $r \in \mathbb{N}_0$ times in $x_{1:n}$, and $m_r := |M_r|$ is their number.

- Assign 3$\times$ uniform probabilities:
  (i) $q_n(x_{1:n}|\mathbf{n}) := \binom{n}{n_1 \ldots n_d}^{-1}$ (as for Laplace)
  (ii) $q_n(\mathbf{n}|\mathbf{m}) := \binom{d}{m_0 \ldots m_n}^{-1}$, where $\mathbf{m} := (m_0, ..., m_n)$
  (iii) $q_n(\mathbf{m}) := \text{Part}(n)^{-1} = (\#\text{integer partitions of } n)^{-1}$

- Together: $q_n(x_{1:n}) = \binom{n}{n_1 \;\ldots\; n_d}^{-1}\binom{d}{m_0 \;\ldots\; m_n}^{-1}\text{Part}(n)^{-1}$ is not TC.

- Normalization: $\tilde{q}^{\text{n}1}(x_{n+1} = i|x_{1:n}) = \frac{1}{\mathcal{N}_n} \cdot \frac{r+1}{n+1} \cdot \frac{m_{r+1}+1}{m_r}$ $[r = n_i]$
  $\mathcal{N}_n := \frac{1}{n+1}\sum_{r=0,m_r\neq 0}^{n}(r+1)(m_{r+1}+1)$

- Is very interesting predictor: $\frac{r+1}{n+1}$ is Laplace is estimate.
  $\frac{m_{r+1}+1}{m_r}$ is close to the Good-Turing (GT) correction $\frac{m_{r+1}}{m_r}$. [Goo53]

Good-Turing (Triple Uniform)

Worst-case regret of GT is very large: $R_n(\tilde{q}^{\text{n1}}||q_n) = n\ln 2 \pm O(\sqrt{n})$

$\Longrightarrow$ Naive norm. severely harms the offline triple uniform estimator q_n

- Heuristic smoothing of the function $m_{()}$
  leads to excellent estimators in practice, [Goo53]
  e.g. Kneser-Ney smoothing for text data. [CG99]
- $\tilde{q}^{\text{mix}}$ may be regarded as an (unusual) kind of smoothing
  with the strong guarantee $R_n \leq 2\ln(n+2)$ [San06]

Ristad (Quadrupel Uniform)

- Motivation: If $\mathcal{X}$ is the set of English words and $x_{1:n}$ some typical English text, then most symbols=words will not appear.

$\Longrightarrow$ Laplace assigns not enough probability ($\frac{n_i+1}{n+d} \ll \frac{n_i}{n}$) to observed words.

- Rectification: Treat symbols $\mathcal{A} := \{i : n_i > 0\}$ that do appear different from symbols $\mathcal{X} \setminus \mathcal{A}$ that don't:

(i) $x_{1:n}$ may contain m different symbols, so $q_n(m) := 1/\min\{n,d\}$

(ii) Choose uniformly which $m \equiv |\mathcal{A}|$ symbols appear: $q_n(\mathcal{A}|m) := \binom{d}{m}^{-1}$

(iii) Choose counts $\mathbf{n}$ ($n_i > 0 \Leftrightarrow i \in \mathcal{A}$) uniformly: $q_n(\mathbf{n}|\mathcal{A}) = \binom{n-1}{m-1}^{-1}$

(iv) Finally, $q_n(x_{1:n}|\mathbf{n}) = \binom{n}{n_1 \dots n_d}^{-1}$ as for Laplace.

- Together: $\quad q_n(x_{1:n}) = \binom{n}{n_1 \; \dots \; n_d}^{-1} \binom{n-1}{m-1}^{-1} \binom{d}{m}^{-1} \frac{1}{\min\{n,d\}} \quad$ is not TC

Ristad (Quadrupel Uniform)

Normalization:

$$\tilde{q}^{\text{n1}}(x_{n+1}=i|x_{1:n}) \;=\; \begin{cases} \frac{(n_i+1)(n-m+1)}{n(n+1)+2m} & \text{if} \quad n_i>0 \text{ and } m<d \\ \frac{m(m+1)}{n(n+1)+2m}\cdot\frac{1}{d-m} & \text{if} \quad n_i=0 \\ \frac{n_i+1}{n+m} & \text{if} \quad m=d \;\; [\Rightarrow n_i>0] \end{cases}$$

Regret of Ristad estimator: $R_n(\tilde{q}^{\text{n1}}||q_n) \leq 2\ln n$

- This shows that simple normalization does not ruin performance.
- Indeed, the regret bound is as excellent as that for $\tilde{q}^{\text{mix}}$

COMPUTATIONAL COMPLEXITY

Computability and Complexity of $\tilde{q}^{\text{mix}}$

- From the four discussed online estimators only $\tilde{q}^{\text{mix}}$ guarantees small extra regret over offline (q_n),
- Problem: The definition of $\tilde{q}^{\text{mix}}$ is quite heavy.
- At least: $\tilde{q}^{\text{mix}}$ can be computed in double-exponential time:

Theorem (Computational Complexity of $\tilde{q}^{\text{mix}}$)

There is an algorithm A that computes $\tilde{q}^{mix}$ (with uniform choice for Q) to accuracy $|A(x_{1:n},\varepsilon)/\tilde{q}^{mix}(x_{1:n})-1|<\varepsilon$ in time $O(|\mathcal{X}|^{\frac{4}{\varepsilon}|\mathcal{X}|^n})$ for all $\varepsilon>0$.

Allows us to:

- compute the predictive distribution $\tilde{q}^{\text{mix}}(x_t|x_{<t})$ to accuracy ε,
- ensures that $A(x_{1:n},\varepsilon)>(1-\varepsilon)\tilde{q}_n^{\text{mix}}(x_{1:n})$,
- hence $R_n(A(x_{1:n},\varepsilon)||q_n)\leq R_n(\tilde{q}_n^{\text{mix}}(x_{1:n})||q_n)+\frac{\varepsilon}{1-\varepsilon}$, and
- approximate normalization $|1-\sum_{x_{1:n}}A(x_{1:n},\varepsilon)|<\varepsilon$.

Computational Complexity: Definitions

- $\mathrm{TIME}(g(n)) :=$ all algs that run in time $O(g(n))$ on inputs of length n
- Algorithms in $\mathrm{E}^c := \mathrm{TIME}(2^{cn})$ run in exponential time.
- $\mathrm{P} := \bigcup_{k=1}^{\infty} \mathrm{TIME}(n^k)$ is the classical class of all algorithms that run in polynomial time (strictly speaking Function-P or FP). [AB09]
- Theorems are stated for binary alphabet $\mathcal{X} = \mathbb{B} = \{0, 1\}$.
  The generalization to arbitrary finite alphabet is trivial.
- 'For all large n' shall mean 'for all but finitely many n', denoted $\forall' n$.

Computational Complexity of General $\tilde{q}$

Theorem (Sub-optimal fast online for fast offline)

For all $r > 0$ *and* $c > 0$ *and* $\varepsilon > 0$

(ii) $\exists (q_s) \in P\ \forall \tilde{q} \in E^c : R_n \geq r \ln n\ \forall' n$ *[e.g. large c and r]*

(iii) $\exists (q_s) \in TIME(s^{r+1+\varepsilon})\ \forall \tilde{q} \in P : R_n \geq r \ln n\ \forall' n$ *[e.g. small c, ε]*

(iv) $\exists (q_s) \in P : \tilde{q}^{mix} \notin E^c$ *[from (ii) and $R_n(\tilde{q}^{mix}) < 3 \ln n$]*

- (iii) implies that there is an offline estimator (q_s) computable in quartic time s^4 on a RAM for which no polynomial-time online estimator $\tilde{q}$ is as good as $\tilde{q}^{\text{mix}}$.
- The slower (q_s) we admit (larger r), the higher the lower bound gets.
- (ii) says that even algorithms for $\tilde{q}$ running in exponential time 2^{cn} cannot achieve logarithmic regret for all $(q_s) \in \text{P}$.
- In particular this implies that (iv) any algorithm for $\tilde{q}^{\text{mix}}$ requires super-exponential time for some $(q_s) \in \text{P}$ on some arguments.

Computational Complexity of General $\tilde{q}$

- $\mathrm{TIME}^o(g(n))$:= all algs with oracle access that run in time $O(g(n))$
- Each oracle call is counted only as one step. Similarly P^o and $\mathrm{E}^{c,o}$.

Theorem (Very poor fast online using offline oracle)

$$\forall\varepsilon > 0\ \exists o \equiv (q_s) \in E^1\ \forall \tilde{q}^o \in E^{\varepsilon/2,o} : R_n(\tilde{q}^o||q_n) \geq (1-\varepsilon)n\ln 2\ \forall' n$$

Or cruder: $\forall\varepsilon > 0\ \exists o \equiv (q_s)\ \forall \tilde{q}^o \in P^o : R_n(\tilde{q}^o||q_n) \geq (1-\varepsilon)n\ln 2\ \forall' n$

- Strength: It rules out even very modest demands on R_n:
  Trivial $R_n \leq n\ln 2$ unimprovable by a fast $\tilde{q}^o$ with (only) oracle access.

- Weakness: Only applies to online $\tilde{q}$ using (q_s) as a black box oracle.
  That is, $\tilde{q}^o(x_{1:n})$ can call $q_s(z_{1:s})$ for any s and $z_{1:s}$
  and receives the correct answer.

Open Problems

Open Problem (Fast online from offline with small extra regret)

Can every polynomial-time offline estimator (q_n) *be converted to a polynomial-time online estimator* $\tilde{q}$ *with small regret* $R_n(\tilde{q}||q_n) \leq \sqrt{n} \forall' n$*? Or weaker:* $\forall (q_n) \in P \exists \tilde{q} \in P : R_n = o(n)$*? Or stronger:* $R_n = O(logn)^2$*?*

- Would reduce finding good online estimators to the apparently easier problem of finding good offline estimators.
- For specific offline (q_n), does there exist efficient $\tilde{q}$ with small R_n?
- A tractable smoothing of the GT estimator with $R_n = O(\ln n)$.
- Are there offline estimators of practical relevance (such as GT) for which no fast online estimator can achieve logarithmic regret?
- Weaken notion of regret to e.g. expected regret $\mathbb{E}[\ln(q_n/\tilde{q})]$.
- Is $R_n = O(\ln n)$ the best one can achieve in general.
- Devise general techniques to upper bound $R_n(\tilde{q}^{\text{n1}}||q_n)$.

References

M. Hutter.
Offline to Online Conversion.
In Proc. ALT'14, volume 8776 of LNAI, pages 230–244, 2014.

I. J. Good.
The population frequencies of species and the estimation of population parameters.
Biometrika, 40(3/4):237–264, 1953.

M. Hutter.
Optimality of universal Bayesian prediction for general loss and alphabet.
Journal of Machine Learning Research, 4:971–1000, 2003.

J. Poland and M. Hutter.
Asymptotics of discrete MDL for online prediction.
IEEE Transactions on Information Theory, 51(11):3780–3795, 2005.

N. Santhanam.
Probability Estimation and Compression Involving Large Alphabets.
PhD thesis, Univerity of California, San Diego, USA, 2006.