

Entropies in On-line Learning

Generalised Entropies and Asymptotic Complexities of Languages

Yuri Kalnishkan

Department of Computer Science
and Computer Learning Research Centre
Royal Holloway, University of London

2008

- the main result of this talk was published in
Y. Kalnishkan, V. Vovk and M. V. Vyugin. Generalised Entropy and Asymptotic Complexities of Languages. In Learning Theory, 20th Annual Conference on Learning Theory, COLT 2007, volume 4539 of Lecture Notes in Computer Science, pages 293-307, Springer 2007.
- I will use it as an excuse to discuss various things in on-line learning...

Talk Outline

1. On-line Prediction
2. Complexities
3. Convex Losses
4. Main Result
5. Proof Sketch

1. On-line Prediction
2. Complexities
3. Convex Losses
4. Main Result
5. Proof Sketch

Protocol

- we try to predict elements of a sequence $\omega_1, \omega_2, \omega_3, \dots \in \Omega$
- we output predictions $\gamma_1, \gamma_2, \gamma_3, \dots \in \Gamma$
- protocol:
 - FOR $t = 1, 2, \dots$
 - (1) \mathfrak{A} chooses a prediction $\gamma_t \in \Gamma$
 - (2) \mathfrak{A} observes the actual outcome $\omega_t \in \Omega$
 - END FOR
- the quality of predictions is measured by a loss function $\lambda(\omega, \gamma)$
 - loss over T trials sums up to the cumulative loss

$$\text{Loss}_{\mathfrak{A}}(\omega_1, \omega_2, \dots, \omega_T) = \sum_{i=1}^T \lambda(\omega_i, \gamma_i)$$

Formalisation

- a *game* \mathfrak{G} is a triple $\langle \Omega, \Gamma, \lambda \rangle$
 - Ω is the *outcome space*
 - Γ is the *prediction space*
 - $\lambda : \Omega \times \Gamma \rightarrow [0, +\infty]$ is the *loss function*
- in this talk
 - $\Omega = \{\omega^{(0)}, \omega^{(1)}, \dots, \omega^{(M-1)}\}$ is finite
 - Γ is compact
 - λ is continuous
- important special case: binary games
 - $\Omega = \mathbb{B} = \{0, 1\}$
 - $\Gamma = [0, 1]$

Examples

- square-loss game: $\Omega = \{0, 1\}, \Gamma = [0, 1],$
 $\lambda(\omega, \gamma) = (\omega - \gamma)^2$
- absolute-loss game: $\Omega = \{0, 1\}, \Gamma = [0, 1],$
 $\lambda(\omega, \gamma) = |\omega - \gamma|$
- logarithmic game: $\Omega = \{0, 1\}, \Gamma = [0, 1]$

$$\lambda(\omega, \gamma) = \begin{cases} -\log_2(1 - \gamma) & \text{if } \omega = 0 \\ -\log_2 \gamma & \text{if } \omega = 1 \end{cases}$$

— can take the value $+\infty$

- simple prediction game: $\Omega = \Gamma = \{0, 1\}$

$$\lambda(\omega, \gamma) = \begin{cases} 0 & \text{if } \omega = \gamma \\ 1 & \text{if } \omega \neq \gamma \end{cases}$$

Prediction Strategy

- $\mathfrak{A} : \Omega^* \rightarrow \Gamma$ maps finite sequences of previous outcomes to predictions
- we can consider various classes of strategies, e.g., computable, polynomially computable etc
 - but the ultimate goal is to study predictability

1. On-line Prediction

2. Complexities

3. Convex Losses

4. Main Result

5. Proof Sketch

- the loss of a strategy \mathfrak{A} on a sequence $\mathbf{x} = (\omega_1, \omega_2, \dots, \omega_n)$ is

$$\text{Loss}_{\mathfrak{A}}(\mathbf{x}) = \sum_{i=1}^n \lambda(\omega_i, \mathfrak{A}(\omega_1, \omega_2, \dots, \omega_{i-1}))$$

- this can be thought of as complexity of \mathbf{x} w.r.t. \mathfrak{A}
- can we define complexity irrespective of \mathfrak{A} ?
 - if we take ‘the best’ \mathfrak{A} , we can consider its loss as ‘intrinsic’ complexity of \mathbf{x}

Diagonalisation Argument

- every strategy is beaten by some other strategy somewhere
- for every sequence there is a strategy that knows it already
 - unless we take computability into account...
- it is not so easy to define complexity

Predictive Complexity (1)

- a solution: *predictive complexity* [Vovk and Watkins, 1998]
 - a class of semi-computable semi-strategies is considered
 - it usually has an optimal element
 - we can define predictive complexity of a sequence up to a constant
 - there can also be versions up to $o(n)$ [Kalnishkan and M. V. Vyugin, 2002]
 - predictive complexity is not computable
- the theory of predictive complexity is very similar to Kolmogorov complexity
 - the logarithmic loss specifies the ‘negative logarithm of Levin’s a priori semimeasure’, which is a variant of Kolmogorov complexity

Predictive Complexity (2)

- various properties of Kolmogorov complexity can be generalised to predictive complexity
- incompressibility property → unpredictability property [Kalnishkan, Vovk and M. V. Vyugin, 2003]
 - most strings cannot be compressed → most strings cannot be predicted
 - the number of those that can decreases exponentially
- a lot of technical issues come up
 - existence question is still partly open
 - computability often creates problems that do not seem essential

Asymptotic Complexity

- let us consider complexity of *languages* (= sets of strings) instead of individual sequences
- let us consider loss per element
- let us consider limits
- we get something like

$$AC(L) = \inf_{\mathfrak{A}} \lim_{n \rightarrow +\infty} \max_{\mathbf{x} \in L \cap \Omega^n} \frac{\text{Loss}_{\mathfrak{A}}(\mathbf{x})}{n}$$

Questions

- so
- $$AC(L) = \inf_{\mathfrak{A}} \lim_{n \rightarrow +\infty} \max_{\mathbf{x} \in L \cap \Omega^n} \frac{\text{Loss}_{\mathfrak{A}}(\mathbf{x})}{n}$$
- what if there are no \mathbf{x} s of length n ?
 - skip that n
 - what if there are no \mathbf{x} s of length n from some length on?
 - no complexity for finite languages
 - what if the limit does not exist?
 - let us consider upper and lower limits instead
 - if the sequence is infinite, we can first take the limit $\lim_{n \rightarrow +\infty}$ along the sequence and then $\sup_{\mathbf{x} \in L}$
 - fine, two more variations of complexity

Finite Sequences

- let $L \subseteq \Omega^*$ (L is a set of finite sequences)
 - let L be infinite
- *upper* (uniform) complexity:

$$\overline{AC}(L) = \inf_{\mathfrak{A}} \limsup_{n \rightarrow +\infty} \max_{\mathbf{x} \in L \cap \Omega^n} \frac{\text{Loss}_{\mathfrak{A}}(\mathbf{x})}{n}$$

- *lower* (uniform) complexity:

$$\underline{AC}(L) = \inf_{\mathfrak{A}} \liminf_{n \rightarrow +\infty} \max_{\mathbf{x} \in L \cap \Omega^n} \frac{\text{Loss}_{\mathfrak{A}}(\mathbf{x})}{n}$$

- in the former definition we assume $\max \emptyset = 0$ and in the later $\max \emptyset = +\infty$

Infinite Sequences

- let $L \subseteq \Omega^\infty$ (L is a set of infinite sequences)
- for a (finite or infinite) \mathbf{x} let $\mathbf{x}|_n$ be the prefix of \mathbf{x} of length n
- we can consider the set of all finite prefixes of all sequences from L ; it has upper and lower complexities; let us call them *upper uniform* complexity $\overline{AC}(L)$ and *lower uniform* complexity $\underline{AC}(L)$
- *upper non-uniform* complexity:

$$\overline{AC}(L) = \inf_{\mathfrak{A}} \sup_{\mathbf{x} \in L} \limsup_{n \rightarrow +\infty} \frac{\text{Loss}_{\mathfrak{A}}(\mathbf{x}|_n)}{n}$$

- *lower non-uniform* complexity:

$$\underline{AC}(L) = \inf_{\mathfrak{A}} \sup_{\mathbf{x} \in L} \liminf_{n \rightarrow +\infty} \frac{\text{Loss}_{\mathfrak{A}}(\mathbf{x}|_n)}{n}$$

Differentiating Examples

- all complexities we have defined are different
- consider a set of infinite sequences with alternating constant and random intervals; it has high upper complexity and low lower complexity
 - at the end of a constant interval the loss per element gets low and at the end of a random interval it gets high
 - the lengths of the intervals should be growing quickly
- consider a set of infinite sequences that have zeros from some point on; it has low non-uniform complexity and high uniform complexity
 - on each sequence loss per element gets low, but this can happen very late

Problem

- suppose we have two loss functions λ_1 and λ_2 that specify complexities AC_1 and AC_2
- what are the relations between AC_1 and AC_2 ?
- a similar question for predictive complexity was addressed in [Kalnishkan 1999, 2002]
- we shall give an answer of the following kind:
 - we shall describe the set of all pairs $\{(AC_1(L), AC_2(L))\}$ on \mathbb{R}^2

1. On-line Prediction
2. Complexities
3. Convex Losses
4. Main Result
5. Proof Sketch

Restriction

- we shall restrict ourselves to one important class of games
- I will touch on the problem of *prediction with expert advice*

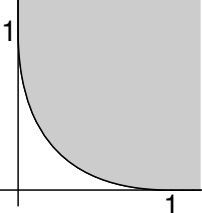
Geometric Interpretation

- consider a game $\langle \Omega, \Gamma, \lambda \rangle$ with finite $\Omega = \{\omega^{(0)}, \omega^{(1)}, \dots, \omega^{(M-1)}\}$
- let us take the set $P = \{(\lambda(\omega^{(0)}, \gamma), \lambda(\omega^{(1)}, \gamma), \dots, \lambda(\omega^{(M-1)}, \gamma)) \mid \gamma \in \Gamma\} \subseteq \mathbb{R}^M$ — images of points from Γ in \mathbb{R}^M
- a point $(s_0, s_1, \dots, s_{M-1}) \in \mathbb{R}^M$ is a *superprediction* if there is $p = (p_0, p_1, \dots, p_{M-1})$ such that

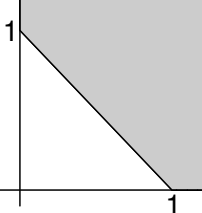
$$\begin{aligned}
 p_0 &\leq s_0 \\
 p_1 &\leq s_1 \\
 &\dots \\
 p_{M-1} &\leq s_{M-1}
 \end{aligned}$$

- superpredictions are located 'above and to the right' from points of P

Examples (1)

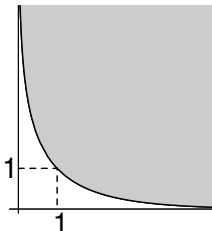


The square-loss game
 $\lambda(\omega, \gamma) = (\omega - \gamma)^2$

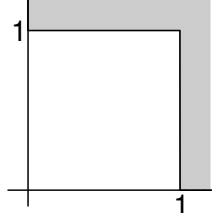


The absolute-loss game
 $\lambda(\omega, \gamma) = |\omega - \gamma|$

Examples (2)



The logarithmic game
 $\lambda(\omega, \gamma) = \begin{cases} -\log_2(1 - \gamma), & \omega = 0 \\ -\log_2 \gamma, & \omega = 1 \end{cases}$



The simple prediction game
 $\lambda(\omega, \gamma) = \begin{cases} 0, & \omega = \gamma \\ 1, & \omega \neq \gamma \end{cases}$

- if the set of superpredictions is convex, we call the game *convex*
 - the square-loss, logarithmic and absolute-loss games are convex
 - the simple prediction game is not
- this geometric property has interesting consequences...

- suppose there are N experts $\mathcal{E}_1, \mathcal{E}_2, \dots, \mathcal{E}_N$ that try to predict elements of the same sequence
 - their predictions become available to us before we make ours
 - we want to predict (nearly) as well as the best expert in terms of cumulative loss
- the amended protocol:
 - (1) FOR $t = 1, 2, \dots$
 - (2) \mathfrak{M} reads predictions $\gamma_t^{(1)}, \gamma_t^{(2)}, \dots, \gamma_t^{(N)} \in \Gamma$
 - (3) \mathfrak{M} chooses a prediction $\gamma_t \in \Gamma$
 - (4) \mathfrak{M} observes the actual outcome $\omega_t \in \Omega$
 - (5) END FOR

Discussion

- we want a merging strategy achieving loss of the kind:

$$\text{Loss}_{\mathfrak{M}} \leq f(\text{Loss}_{\mathcal{E}_i})$$

where \mathcal{E}_i is the best expert so far

- no limitations are imposed on experts
 - they do not have to be computable etc
 - in fact, they are just metaphors for series of predictions appearing in the protocol
 - this is a game of the ‘learner’ vs ‘nature plus experts’
- the problem has been studied since 1980s; a good overview can be found in

Prediction, learning, and games, Nicolò Cesa-Bianchi and Gábor Lugosi, Cambridge University Press, 2006

Weak Mixability

- it was shown in [Kalnishkan and M. V. Vyugin, 2005] that for convex games and only for them there is a strategy achieving loss that satisfies

$$\text{Loss}_{\mathfrak{M}}(\mathbf{x}) \leq \text{Loss}_{\mathcal{E}_i}(\mathbf{x}) + o(|\mathbf{x}|)$$

for every sequence \mathbf{x} and every expert \mathcal{E}_i (here $|\mathbf{x}|$ is the length of \mathbf{x})

- if, moreover, the loss function is bounded, $o(|\mathbf{x}|)$ can be replaced by $O(\sqrt{|\mathbf{x}|})$
- the result about the square root was proved independently in [Hutter and Poland, 2005] a bit earlier
 - we used a weighted majority-type algorithm and they used an algorithm from following the perturbed leader family

Mixability

- and BTW the concept of *weak mixability* originated from the concept of *mixability*
- let us apply the transformation $x \rightarrow e^{-\eta x}$ to all coordinates of the set of superprediction
- there is $\eta > 0$ such that the image of S under this transformation is convex if and only if we can have a constant extra term [Vovk 1991, 1998]

1. On-line Prediction

2. Complexities

3. Convex Losses

4. Main Result

5. Proof Sketch

Entropy

- let p^* be a probability distribution on Ω
 - $p^* = (p_0, p_1, \dots, p_{M-1})$, where $\sum p_i = 1$
- *generalised entropy*

$$H(p^*) = \min_{\gamma \in \Gamma} \mathbf{E}_{p^*} \lambda(\omega, \gamma) = \min_{\gamma \in \Gamma} \sum_{i=0}^{M-1} p_i \lambda(\omega^{(i)}, \gamma)$$

- suppose we know that the next outcome is distributed according to P^*
- we will be looking for $\gamma \in \Gamma$ to minimise the expected loss
- the minimum of the expected loss is the entropy $H(p^*)$

Binary Case

- in the binary case let p be the probability of 1
 - then $(1 - p)$ is the probability of 0
 - a distribution can be identified with $p \in [0, 1]$
- entropy is given by

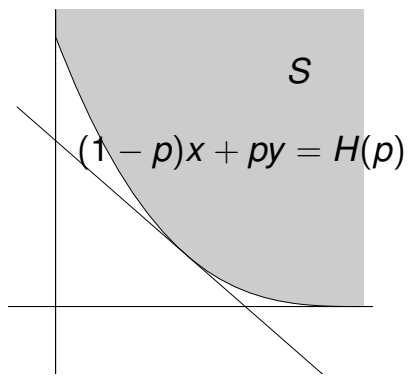
$$H(p) = \min_{\gamma \in [0,1]} [(1 - p)\lambda(0, \gamma) + p\lambda(1, \gamma)]$$

- for the logarithmic game

$$H(p) = \min_{\gamma \in [0,1]} [-(1 - p) \log(1 - \gamma) - p \log \gamma]$$

- one can check (eg, by differentiation) that min is achieved on $\gamma = p$
- thus $H(p) = -(1 - p) \log(1 - p) - p \log p$
- this is Shannon entropy

Geometric Interpretation



- the entropy corresponds to the tangent line with a given slope

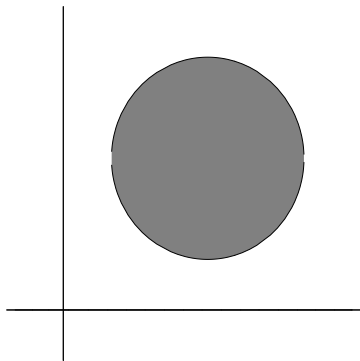
Entropy Hull

- suppose we have two games \mathcal{G}_1 and \mathcal{G}_2 (with the same Ω) — they specify two entropies H_1 and H_2
- consider the set $\{(H_1(p^*), H_2(p^*)) \mid p^* \text{ is a distribution}\}$
- $\mathcal{G}_1/\mathcal{G}_2$ -entropy hull is its convex hull
- this is nearly the solution to our problem...

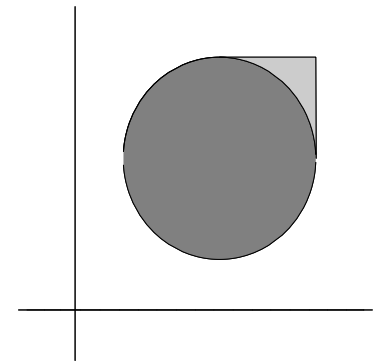
Some Planar Sets

- we say that a convex set $M \subseteq \mathbb{R}^2$ is a *spaceship* if
 - for every two points $(x_1, y_1), (x_2, y_2) \in M$ we have
 - the point $(\max(x_1, x_2), \max(y_1, y_2)) \in M$
- a convex set that is not a spaceship is a *turnip*
- the *spaceship closure* of a set is the minimal spaceship containing the set

Example



a turnip T

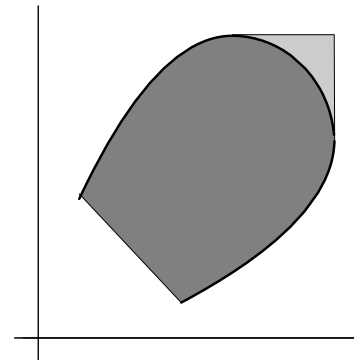


the spaceship closure of T

Main Theorem

- let \mathcal{G}_1 and \mathcal{G}_2 be games with the same set of outcomes and convex loss functions
- then the set of all pairs $(AC_1(L), AC_2(L))$, where
 - AC is any of the complexities \overline{AC} , \underline{AC} , $\overline{\overline{AC}}$, or $\underline{\underline{AC}}$
 - L ranges over all non-empty sets of infinite sequences or all infinite sets of finite sequences accordingly
- coincides with the spaceship closure of the $\mathcal{G}_1/\mathcal{G}_2$ -entropy hull

Building a Spaceship



entropy curve \rightarrow entropy hull \rightarrow spaceship

Discussion

- the requirement of convexity cannot be omitted; indeed, consider the simple prediction game
 - $H(p) = \min(p, 1 - p) \leq 1/2$
 - $AC(\mathbb{B}^*) = 1$ (due to the diagonalisation argument)
- AC_1 and AC_2 must be the same types of complexity; indeed, let $\mathcal{G}_2 = \mathcal{G}_1$
 - the $\mathcal{G}_1/\mathcal{G}_1$ -entropy hull is a subset of $x = y$
 - and we know that complexities differ

1. On-line Prediction

2. Complexities

3. Convex Losses

4. Main Result

5. Proof Sketch

Complexities Belong to the Spaceship Closure

Proof of the Lemma

- the proof is based on the *recalibration lemma*
- suppose we have a prediction strategy \mathfrak{A} for \mathcal{G}_1
- then we can construct \mathfrak{A}_1 for \mathcal{G}_1 and \mathfrak{A}_2 for \mathcal{G}_2 so that
 - for every finite sequence \mathbf{x} there is $(u_{\mathbf{x}}, v_{\mathbf{x}})$ from the $\mathcal{G}_1/\mathcal{G}_1$ -entropy hull such that

$$\text{Loss}_{\mathfrak{A}_1}^{\mathcal{G}_1}(\mathbf{x}) \lesssim u_{\mathbf{x}}|\mathbf{x}|$$

$$\text{Loss}_{\mathfrak{A}_2}^{\mathcal{G}_2}(\mathbf{x}) \lesssim v_{\mathbf{x}}|\mathbf{x}|$$

— but still \mathfrak{A}_1 is an improvement on \mathfrak{A} :

$$u_{\mathbf{x}}|\mathbf{x}| \lesssim \text{Loss}_{\mathfrak{A}}^{\mathcal{G}_1}(\mathbf{x})$$

- the strategy \mathfrak{A} can be discretised at a small cost
- the discretised strategy identifies finitely many ‘cases’ and makes a prediction for each
 - perhaps it makes wrong predictions; let us change those
 - we do not know how
 - let us take all possible shuffles of the finite set of predictions; we shall obtain a new prediction strategy for each
 - let us use convexity and aggregate them all

Complexities Fill the Spaceship Closure

Acknowledgements

- a basic building block is the set of strings of length n that have approximately $p_i n$ elements equal to $\omega^{(i)}$ (give or take)
- these building blocks can be used to construct languages of complexities $H_1(p^*)$ and $H_2(p^*)$
- combining these blocks allows us to fill the entropy hull
- taking the union allows us to fill the starship closure

- I am grateful to Ivan Polikarov of Moscow State University for his inspiring ideas on turnips and spaceships.