Complexity, Stability, and Robustness: A Unified Perspective on Predictive System Behavior

Università degli Studi dell'Aquila
September 22, 2025

Joel Ratsaby
Ariel University

Overview

Central Question:

What is the relationship between *complexity*, *stability* and *performance* of a system?

Examples: Ecological Systems (May, 1974): the *more* complex a system, the *less* stable it becomes. In *Software Engineering*, the *reliability* of software is directly affected by its complexity (Lew et al., 1988)

Objective: investigate how the following system properties *interrelate*:
– complexity
– stability
– performance guarantee

Notion of a *system* can vary broadly across different areas of science and engineering.

Scope of Analysis

Systems that *predict* binary Markov chains

Analysis *demonstrates* the relationships between these three concepts.

Build on principles from *mathematical statistics*, *information theory*

Environment

Markov chain (discrete, stationary, homogenous)

$$\{X_t : t \in \mathbb{Z}\},\ X_t \in \{-1, 1\}$$

$$P\left(X_t = x \mid \underbrace{X_{t-1} = x_{t-1}, \ldots, X_{t-k^*} = x_{t-k^*}}_{S^*_{t-\mathbf{1}}}\right)$$

Order $k^* \in \mathbb{Z}_{\geq 1}$ (unknown).

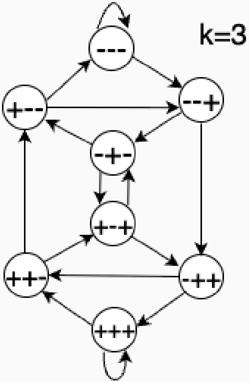
Transition probabilities $P(X_t = 1 \big| S^*_{t-1} = s)$, $P(X_t = -1 \big| S^*_{t-1} = s)$ (unknown).

State Space

State Space

$\mathbb{S}_k$ consists of states $s \in \{-1, 1\}^k$, k may differ from k^* (because k^* unknown).

Definition: State transitions are based on de Bruin graph† of dimension k.

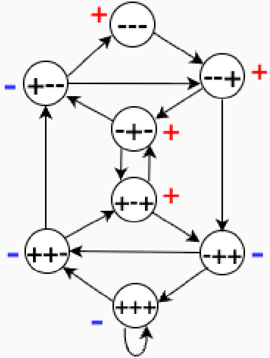


† de Bruijn (1946)

Classifier

Binary function on $\mathbb{S}_k$,

$$h : \mathbb{S}_k \to \{-1, 1\}$$



Distance

Definition: $\mathrm{d}(s, s')$ the *distance* between states s and s' is the *length* of the *shortest* path between their corresponding nodes on the *undirected* graph.

Definition: $\operatorname{diam} \mathbb{S}_k := \max_{s,s' \in \mathbb{S}_k} \mathrm{d}(s, s')$.

We have, $\operatorname{diam} \mathbb{S}_k = k$.

Definition: For $R \subseteq \mathbb{S}_k$, let

$$\operatorname{dist}(s, R) := \min_{s' \in R} \mathrm{d}(s, s')$$

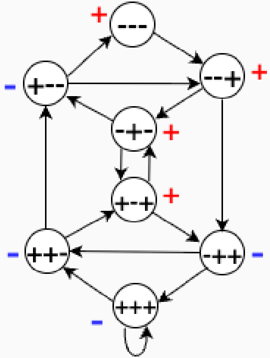
Width & Margin

Width of classifier h

For $s \in \mathbb{S}_k$, the width of h at s is

$$w_h(s) := \operatorname{dist}\left(s, R_{\overline{h}(s)}\right)$$

where R_+, $R_- \subseteq \mathbb{S}_k$ are regions classified as 1 and -1 by h, and $\overline{h}(s)$ is the complement of $h(s)$.



Margin of classifier h

For $s \in \mathbb{S}_k$, the *margin* of h at s is

$$f_h(s) := h(s) w_h(s)$$

Absolute value $|f_h(s)|$ gives indication of the *confidence* in the decision of $h(s)$.

Example: Suppose $h(s) = h(s') = -1$ and $f_h(s) = -5$, $f_h(s') = -2$, then we are more *confident* in h's decision for s than for s'.

Prediction System

Confident Decision

Use h to decide *only* if *sufficient* confidence

Input: State $S_{t-1} := \left(X_{t-k}, X_{t-(k-1)}, \ldots, X_{t-1}\right) \in \mathbb{S}_k$

Let $\gamma \in (0, k]$ be a parameter, *confidence threshold* $a(\gamma) \geq 0$ be nondecreasing, with $a(0) = 0$.

System $:= (h, \gamma)$

$$\text{Predict } X_t = \begin{cases} 1 & \text{if } f_h(S_{t-1}) \geq a(\gamma) \\ -1 & \text{if } f_h(S_{t-1}) \leq -a(\gamma) \\ \text{reject making decision.} & \text{otherwise} \end{cases}$$

System predicts whenever the margin *exceeds* the confidence threshold $|f_h(S_{t-1})| \geq a(\gamma)$.

No Assumption

on how a system is produced, for instance, it can be defined based on prior knowledge or learned from data, or any other way

We define the output of the system:

Output: $Y_t \in \{-1, 1\}$ equals 1 if the system predicts *wrongly*, $h(S_{t-1}) \neq X_t$, or -1 if the system predicts *correctly*

It can be expressed as: $Y_t := \begin{cases} X_t & \text{if } f_h(S_{t-1}) \leq -a(\gamma) \\ \overline{X}_t & \text{if } f_h(S_{t-1}) \geq a(\gamma) \\ \text{null} & \text{otherwise} \end{cases}$

where $\overline{X} := \begin{cases} 1 & \text{if } X = -1 \\ -1 & \text{if } X = 1 \end{cases}$ and null means no output.

Remark

At any time t, the system acts as *switch*: it *decides* whether to *select* an input bit (or its complement). Then, if it selects a bit, it *places* it as an output bit

Error

Prediction Error

The event, $h(S_{t-1}) \neq X_t$, can be expressed in terms of the margin function as

$$X_t f_h(S_{t-1}) < 0$$

Let *margin penalty* $b(\gamma) \geq 0$ be non-decreasing function with $b(0) = 0$.

Margin Error

$$X_t f_h(S_{t-1}) < b(\gamma)$$

A margin error occurs if the prediction is *incorrect* or if it is *correct* and has a *low* confidence.

Error Sequence

Input: $X^{(n)}$, sample of consecutive bits from the environment

System: (h, γ) aims to predict $X^{(n)}$

Denote $\nu := \nu^{(a)} \leq n$ the number of times that system makes a prediction.

Output: $Y^{(\nu)} := \{Y_{t_l}\}_{l=1}^{\nu^{(a)}}$

We assess the system by observing its prediction errors.

Error Sequence

$$\Psi^{(\nu^{(a)})}(h) := \{\Psi_{t_l}\}_{l=1}^{\nu^{(a)}} = \left\{\mathbb{I}\left\{X_{t_l} f_h(S_{t_l-1}) < 0\right\}\right\}_{l=1}^{\nu^{(a)}}$$

$\Psi_{t_l} = \varphi\left(Y_{t_l}\right)$, where $\phi(y) := \frac{y+1}{2}$ so the output sequence $Y^{(\nu)}$ contains all the information about the system's prediction errors.

Average Prediction Error

$$\mathfrak{L}_{\nu}^{(n,\gamma)}(h) := \frac{1}{\nu} \sum_{l:\left|f_h(S_{t_l-\mathbf{1}})\right| \geq a(\gamma)} \Psi_{t_l}$$

Margin Error Sequence

Input: $X^{(m)}$, a second sample of the environment

Let $\gamma = 0$, confidence threshold $a(0) = 0$ (system never rejects) and $\nu := \nu^{(0)} = m$.

System: $(h, 0)$ aims to predict $X^{(m)}$

Output: $Y^{(m)} := \{Y_t\}_{t=1}^{m}$

We assess the system *more strictly* by observing its margin errors.

Margin Error Sequence

$$\Psi^{(m,\gamma)}(h) := \left\{\Psi_t^{(m,\gamma)}(h)\right\}_{t=1}^{m} = \left\{\mathbb{I}\left\{X_t f_h(S_{t-1}) < b(\gamma)\right\}\right\}_{t=1}^{m}$$

Penalize the system even when it predicts *correctly* with an insufficient level b of confidence.

Average Margin Error

$$L_m^{(b(\gamma))}(h) := \frac{1}{m} \sum_{t=1}^{m} \Psi_t^{(m,\gamma)}(h)$$

Discrepancy

Discrepancy

$$\Upsilon_{m,n}(h,\gamma) := \mathfrak{L}_{\nu^{(a)}}^{(n,\gamma/2)}(h) - L_m^{(b(2\gamma))}(h)$$

Discrepancy measures the *difference* in performance between two systems that have the *same* classifier h.

One system is based on a *positive* decision confidence threshold $a(\gamma) > 0$ whose performance is measured on sample $X^{(n)}$ with penalty *only* if wrong prediction.

The other is based on a *zero* decision confidence threshold with performance based on $X^{(m)}$ and *higher* penalty (margin error).

Admissibility

Definition: Probability of *false* prediction at time t by system $(h, 0)$

$$p_0 := P\left(X_t f_h\left(S_{t-1}\right) < 0\right)$$

Definition: Probability of *false* prediction at time t by system (h, γ)

$$p_a := P\left(X_t f_h\left(S_{t-1}\right) < 0 \middle| \left|f_h\left(S_{t-1}\right)\right| \geq a(\gamma)\right)$$

Admissible System

$$p_a \leq p_0$$

Using the *same* classifier h with a confidence threshold that is strictly *larger* than zero *cannot* worsen the classification.

Admissibility is satisfied by any *reasonably* good system.

Complexity

Definition[†]: System complexity is the *uncertainty* that a system meets its functional requirements.

System Complexity[‡]

Let (h, γ) be a prediction system with confidence function $a(\gamma)$. Let $X^{(n)}$ be a sample input sequence from the environment and $Y^{(\nu)}$ the corresponding system's output sequence. The system's complexity is

$$\mathcal{C}(h, \gamma) := \frac{1}{n} H\left(Y^{(\nu)} \,|\nu\right)$$

$H(Y^{(\nu)}|\nu)$ is conditional entropy of the sequence $Y^{(\nu)}$ given its length ν.

System complexity

Average number of information bits (minimal expected *description length*) per input bit, for describing failures of a system in predicting a sample of the environment

[†] Suh (2005), *Complexity: Theory and Applications*, Oxford University Press

[‡] Ratsaby (2024)

Definition: Probability of making a prediction,

$$\mathrm{P}_a := \mathrm{P}_a^{(h,\gamma)} := P\left(\left|f_h\left(S_{t-1}\right)\right| \geq a(\gamma)\right)$$

Lemma 1

The complexity of system (h, γ) is bounded as

$$0 \leq \mathcal{C}(h, \gamma) \leq H\left(p_a\right) \mathrm{P}_a \leq 1$$

where $H(p) := -p \log p - (1-p) \log(1-p)$ is binary entropy and p_a is probability of error.

Remark

As $\gamma \nearrow$, $\mathrm{P}_a \searrow$ since $a(\gamma)$ is non-decreasing, the *error* probability $p_a \searrow$ since the system is admissible, $H(p_a) \searrow$ hence $\mathcal{C}(h, \gamma) \searrow$

Stability

Null Hypothesis: For an admissible system (h, γ), the expected discrepancy $\mathbb{E}\Upsilon_{m,n}(h, \gamma) \leq 0$.

It can be shown that the Null hypothesis *holds*.

Draw two samples $X^{(n)}$, $X^{(m)}$ from the environment and evaluate the discrepancy $\Upsilon_{m,n}(h, \gamma)$.

Critical Value: For any $0 < \delta < 1$, let

$$\epsilon := \epsilon(m, n, \gamma, \delta) = O\left(\sqrt{\tfrac{1}{\min\{n,m\}} N_\gamma \ln \tfrac{k}{\gamma\delta}}\right)$$

where N_γ is covering number of $\mathbb{S}_k$ with respect to distance d.

Significance Test: If $\Upsilon_{m,n}(h, \gamma) > \epsilon$ then reject the Null hypothesis.

If *Reject* the Null hypothesis then declare system *unstable* (similar to Statistical Process Control) otherwise we declare it ϵ-*stable*.

Theorem 2

Probability is at most δ that there exists an *admissible* system (h, γ) with discrepancy $\Upsilon_{m,n}(h, \gamma) > \epsilon$

This means that we wrongly *declare* a system *unstable* (due to discrepancy $> \epsilon$) with *no more* than δ probability.

As $\gamma \nearrow$, the critical value $\epsilon(m, n, \gamma, \delta) \searrow$ therefore with Lemma 1 we have:

Remark

A *less* complex system is *more* stable (has a smaller ϵ)

Performance

Input: $X^{(m)}$, sample of the environment

Definition: *system performance guarantee*: upper bound on the prediction error p_a.

Use the margin error sequence $\Psi^{(m,\gamma)}(h)$ but consider just the times t where the margin $|f_h(S_{t-1})| \geq a(\gamma/6)$.

Average Prediction Margin Error

$$\mathcal{L}_{\nu^{(a)}}^{(m,\gamma/6)}(h) := \frac{1}{\nu^{(a)}} \sum_{l:\left|f_h(S_{t_l-1})\right| \geq a(\gamma/6)} \Psi_{t_l}^{(m,\gamma)}$$

Let $\xi(m,\gamma,\delta) = O\left(\sqrt{\frac{1}{m} N_\gamma \ln \frac{k}{\gamma\delta}}\right)$.

Performance Guarantee Function:

$$\hat{\mathcal{L}}_m^{(\gamma,\delta)}(h) := \mathcal{L}_{\nu^{(a)}}^{(m,\gamma/6)}(h) + \xi(m,\gamma,\delta)$$

Theorem 3

Probability is at most δ that there exists a system (h, γ) such that its error $p_a > \hat{\mathcal{L}}_m^{(\gamma,\delta)}(h)$

With confidence at least $1 - \delta$, the probability of system's failure (error probability) is *no* larger than the performance guarantee function (therefore it serves as system performance guarantee).

Remark

System with a large γ can be assessed *more* accurately. For instance, suppose $\mathcal{L}_{\nu^{(a)}}^{(m,\gamma/6)}(h) \simeq 0$. Then, since $\xi(m, \gamma, \delta)$ is small (due to a large γ) its prediction error p_a is small, with large confidence. This cannot be said for a system with a small value of γ.

Remark

γ influences the *sensitivity* of the performance guarantee function to the input $X^{(m)}$. A *higher* value of γ means the performance guarantee function is *less* sensitive to a change in the input, therefore the system is *more* robust to the *randomness* of the environment.

Conclusions

Prediction system (h, γ): binary classifier h and *confidence* threshold $a(\gamma)$.

Acts like a *switch* that *selects* bits (or their complements) from the input and copies them as output bits.

Output: Binary sequence that indicates system's prediction *failures*.

Complexity: $\mathcal{C}(h, \gamma)$ is average minimal number of information bits needed to describe the output, per input bit.

Stability: having a small discrepancy $\Upsilon_{m,n}(h, \gamma) \leq \epsilon$.

Performance guarantee: an upper *bound* on the prediction error probability.

Summary of the Results

As the system's complexity $\mathcal{C}(h, \gamma)$ *decreases*, the system becomes *more* stable, can be assessed *more* accurately, and has a *more* robust performance guarantee (less sensitive to changes in *random* input from its environment).

Thanks for your attention.