Online nearest neighbor classification

Sanjoy Dasgupta and Geelon So (2023)

Geelon So, agso@eng.ucsd.edu
Research Exam — Aug 28, 2023

Weather forecasting problem

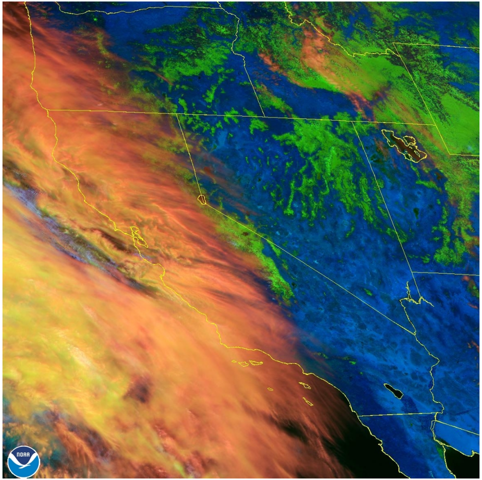
THE WEATHER CHANNEL'S TASK

Weather forecasting problem

THE WEATHER CHANNEL'S TASK

Each day:

- receive current atmospheric data

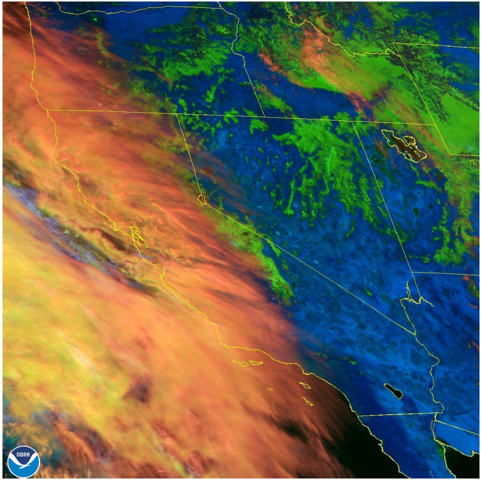


Weather forecasting problem

THE WEATHER CHANNEL'S TASK

Each day:

- **receive** current atmospheric data
- **predict** tomorrow's weather

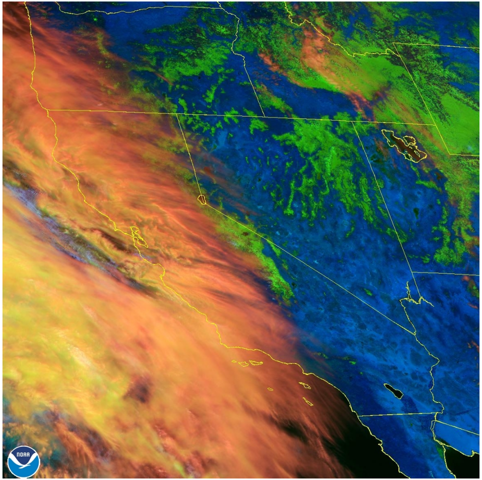


Weather forecasting problem

THE WEATHER CHANNEL'S TASK

Each day:

- **receive** current atmospheric data
- **predict** tomorrow's weather
- **observe** actual weather

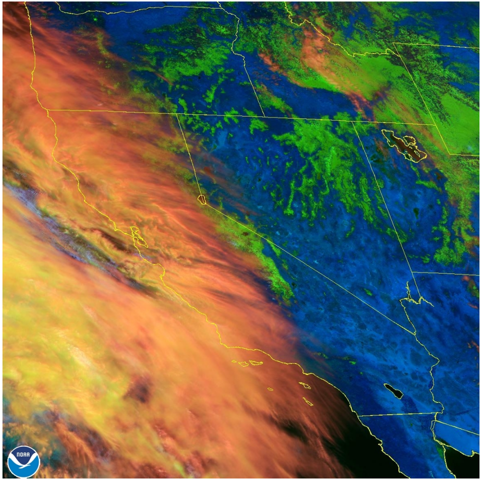


Weather forecasting problem

THE WEATHER CHANNEL'S TASK

Each day:

- **receive** current atmospheric data
- **predict** tomorrow's weather
- **observe** actual weather
- **incur** ire of viewers if wrong



Nearest neighbor for weather prediction

NEAREST NEIGHBOR ALGORITHM

- remember all past conditions + weather outcomes

Nearest neighbor for weather prediction

NEAREST NEIGHBOR ALGORITHM

- remember all past conditions + weather outcomes
- predict weather according to the most similar conditions in memory

The nearest neighbor rule

SETTING

Let $(\mathcal{X}, \rho)$ be a metric space.

The nearest neighbor rule

SETTING

Let $(\mathcal{X}, \rho)$ be a metric space.

NEAREST NEIGHBOR ALGORITHM

The nearest neighbor rule

SETTING

Let $(\mathcal{X}, \rho)$ be a metric space.

NEAREST NEIGHBOR ALGORITHM

- remember all past data points $\{(x_1, y_1), \ldots, (x_t, y_t)\}$

The nearest neighbor rule

SETTING

Let $(\mathcal{X}, \rho)$ be a metric space.

NEAREST NEIGHBOR ALGORITHM

- remember all past data points $\{(x_1, y_1), \ldots, (x_t, y_t)\}$
- given query x, find most similar data point in memory

$$\mathrm{NN}(x) = \arg\min_{\tau} \rho(x, x_\tau)$$

The nearest neighbor rule

SETTING

Let $(\mathcal{X}, \rho)$ be a metric space.

NEAREST NEIGHBOR ALGORITHM

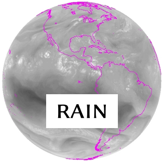
- remember all past data points $\{(x_1, y_1), \ldots, (x_t, y_t)\}$
- given query x, find most similar data point in memory

$$\mathrm{NN}(x) = \arg\min_{\tau} \rho(x, x_\tau)$$

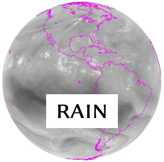
- predict using corresponding label

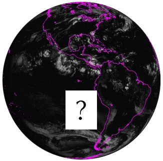
$$\hat{y}(x) = y_{\mathrm{NN}(x)}$$

Nearest neighbor for weather prediction

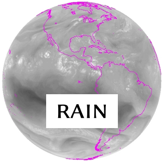


Nearest neighbor for weather prediction

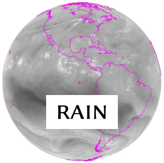


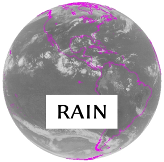


Nearest neighbor for weather prediction

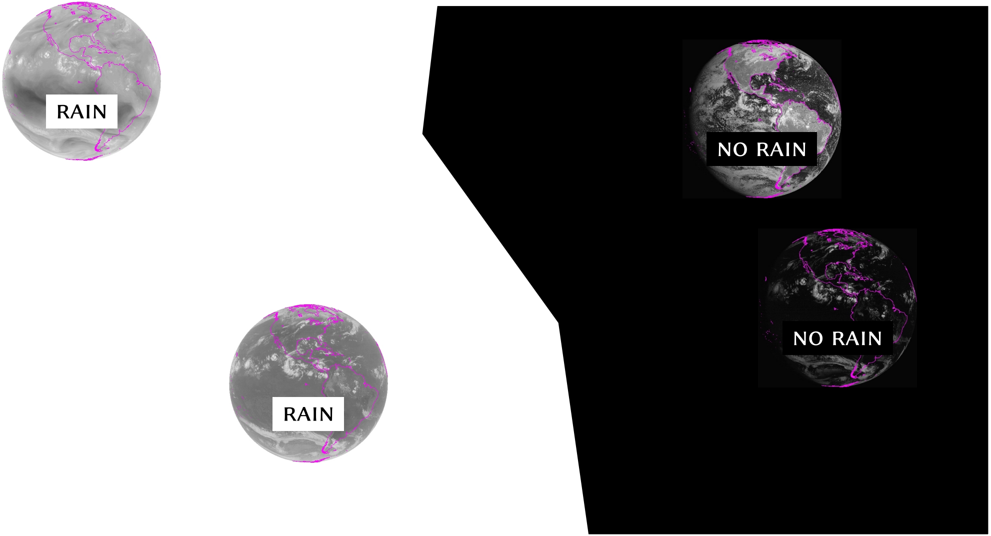


Nearest neighbor for weather prediction

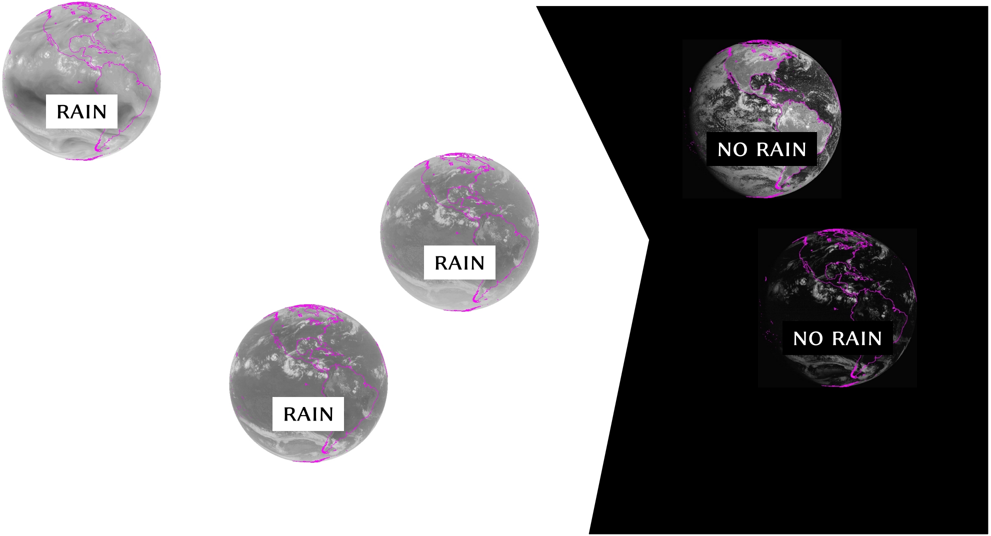




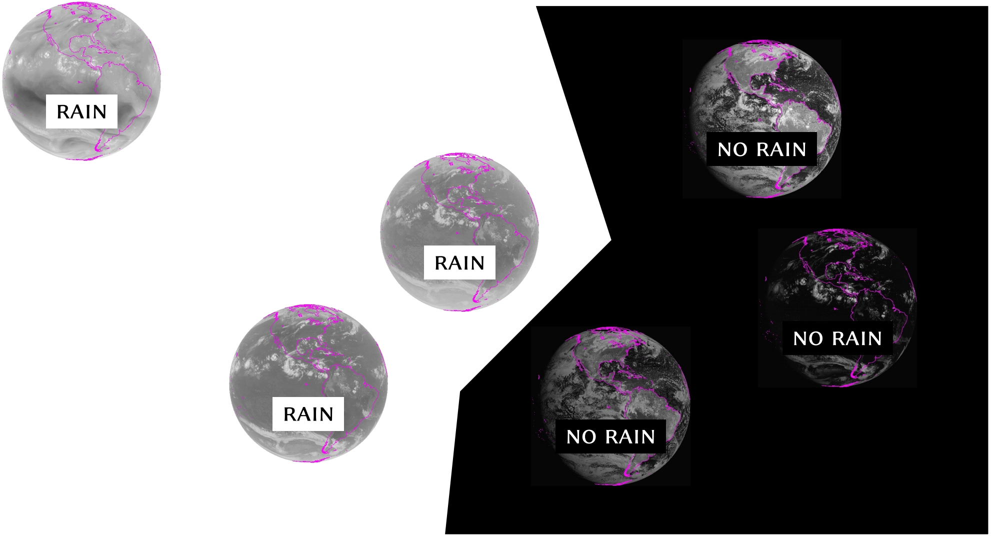
Nearest neighbor for weather prediction



Nearest neighbor for weather prediction



Nearest neighbor for weather prediction



Behavior of online nearest neighbor

QUESTION

When is the *nearest neighbor rule* a reasonable online prediction strategy?

Online learning setting

ONLINE LEARNING LOOP

For $t = 1, 2, \ldots$

- receive instance x_t

Online learning setting

ONLINE LEARNING LOOP

For $t = 1, 2, \ldots$

- receive instance x_t
- predict label $\hat{y}_t$

Online learning setting

ONLINE LEARNING LOOP

For $t = 1, 2, \ldots$

- receive instance x_t
- predict label $\hat{y}_t$
- observe true label y_t

Online learning setting

ONLINE LEARNING LOOP

For $t = 1, 2, \ldots$

- receive instance x_t
- predict label $\hat{y}_t$
- observe true label y_t
- incur loss $\ell(x_t, y_t, \hat{y}_t)$

Online learning setting

REALIZABILITY ASSUMPTION

The true labels are generated by some underlying function $f : \mathcal{X} \to \mathcal{Y}$,

$$y_t = f(x_t).$$

Online learning setting

GOAL

Make fewer and fewer mistakes over time.

Online learning setting

GOAL

Make fewer and fewer mistakes over time. Formally:

$$\underbrace{\mathrm{er}_T := \frac{1}{T}\sum_{t=1}^{T} \ell\big(x_t, y_t, \hat{y}_t\big) \to 0\,.}_{\text{achieve vanishing error rate}}$$

Connection to regret

In the usual goal in the online learning setting is to achieve sublinear regret:

$$\mathrm{regret}_T := \sum_{t=1}^{T} \ell\big(x_t, y_t, \hat{y}_t\big) - \inf_{h \in \mathcal{H}} \sum_{t=1}^{T} \ell\big(x_t, y_t, h(x_t)\big).$$

Connection to regret

In the usual goal in the online learning setting is to achieve sublinear regret:

$$\text{regret}_T := \sum_{t=1}^{T} \ell\big(x_t, y_t, \hat{y}_t\big) - \inf_{h \in \mathcal{H}} \sum_{t=1}^{T} \ell\big(x_t, y_t, h(x_t)\big).$$

- In the realizable setting, if $\mathcal{H}$ is non-parametric (e.g. all nearest neighbor classifiers), no mistakes are made by any optimal $h \in \mathcal{H}$ on $(x_1, y_1), \ldots, (x_T, y_T)$.

Connection to regret

In the usual goal in the online learning setting is to achieve sublinear regret:

$$\mathrm{regret}_T := \sum_{t=1}^{T} \ell\big(x_t, y_t, \hat{y}_t\big) - \inf_{h \in \mathcal{H}} \sum_{t=1}^{T} \ell\big(x_t, y_t, h(x_t)\big).$$

- In the realizable setting, if $\mathcal{H}$ is non-parametric (e.g. all nearest neighbor classifiers), no mistakes are made by any optimal $h \in \mathcal{H}$ on $(x_1, y_1), \dots, (x_T, y_T)$.
- Thus, sublinear regret is equivalent to vanishing error rate.

Difficulty of realizable online learning

- The sequence of instances x_t do not come i.i.d. from some distribution.

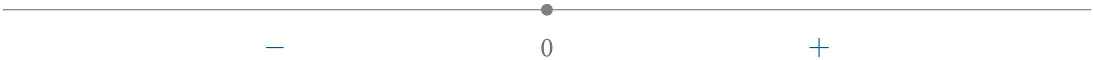
Difficulty of realizable online learning

- The sequence of instances x_t do not come i.i.d. from some distribution.
- In the worst-case, each x_t is selected so that learner makes a mistake each time.

Negative example: learning the sign function

GOAL

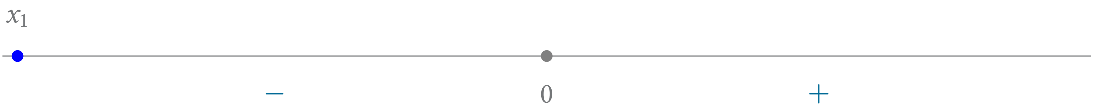
Learn the sign function $\quad f(x) := \begin{cases} + & x \geq 0 \\ - & x < 0 \end{cases}$



Negative example: learning the sign function

GOAL

Learn the sign function $\quad f(x) := \begin{cases} + & x \geq 0 \\ - & x < 0 \end{cases}$

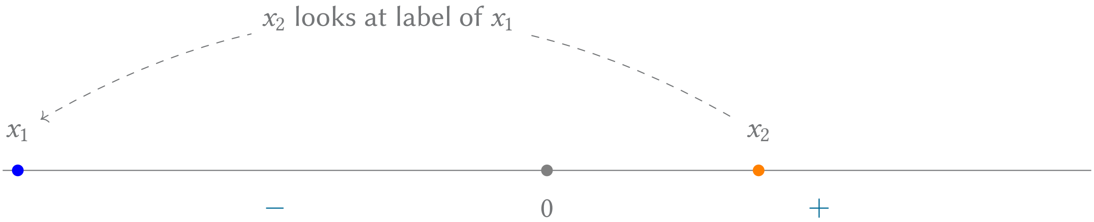


EXAMPLE. A worst-case sequence where the nearest neighbor rule errs every time.

Negative example: learning the sign function

GOAL

Learn the sign function $f(x) := \begin{cases} + & x \geq 0 \\ - & x < 0 \end{cases}$

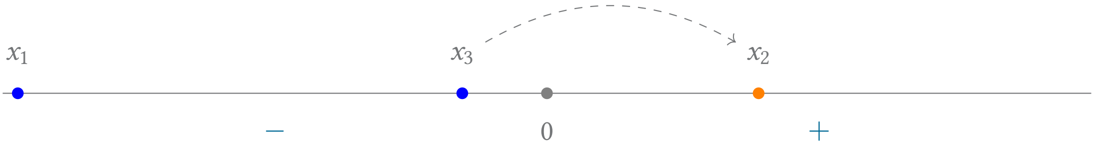


EXAMPLE. A worst-case sequence where the nearest neighbor rule errs every time.

Negative example: learning the sign function

GOAL

Learn the sign function $\quad f(x) := \begin{cases} + & x \geq 0 \\ - & x < 0 \end{cases}$



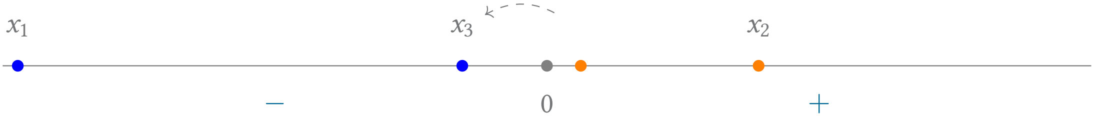
EXAMPLE. A worst-case sequence where the nearest neighbor rule errs every time.

- The sequence alternate signs and the nearest neighbor of x_{t+1} is x_t out of $x_1, \ldots, x_t$.

Negative example: learning the sign function

GOAL

Learn the sign function $f(x) := \begin{cases} + & x \geq 0 \\ - & x < 0 \end{cases}$



EXAMPLE. A worst-case sequence where the nearest neighbor rule errs every time.

- The sequence alternate signs and the nearest neighbor of x_{t+1} is x_t out of $x_1, \ldots, x_t$.
- Mistake rate fails to go to zero despite the mistake set shrinking exponentially fast.

Generalized negative result

SETTING

Let $(\mathcal{X}, \rho)$ be a totally bounded metric space and $f : \mathcal{X} \to \{-, +\}$.

Proposition (Non-convergence in the worst-case)

There is a sequence of instances $(x_t)_t$ on which the nearest neighbor error rate is bounded away from zero if and only if there is no positive separation between classes:

$$\inf_{f(x) \neq f(x')} \rho(x, x') = 0.$$

- Proof idea: can always find arbitrarily close pairs (x, x') with opposite signs
  - can select sequence so that x_{2t} is closest to x_{2t-1}, which has the opposite sign

Implications of negative result

The worst-case adversary is too powerful—learning may not be possible in this setting.

- ISSUE: the adversary can compute/construct test instances with arbitrary precision.

Implications of negative result

The worst-case adversary is too powerful—learning may not be possible in this setting.

- ISSUE: the adversary can compute/construct test instances with arbitrary precision.

Saving grace: we may reasonably never expect to meet this worst-case adversary.

Implications of negative result

The worst-case adversary is too powerful—learning may not be possible in this setting.

- ISSUE: the adversary can compute/construct test instances with arbitrary precision.

Saving grace: we may reasonably never expect to meet this worst-case adversary.

- a real adversary may be limited in:
  - information,

Implications of negative result

The worst-case adversary is too powerful—learning may not be possible in this setting.

- ISSUE: the adversary can compute/construct test instances with arbitrary precision.

Saving grace: we may reasonably never expect to meet this worst-case adversary.

- a real adversary may be limited in:
  - information,
  - computational power,

Implications of negative result

The worst-case adversary is too powerful—learning may not be possible in this setting.

- ISSUE: the adversary can compute/construct test instances with arbitrary precision.

Saving grace: we may reasonably never expect to meet this worst-case adversary.

- a real adversary may be limited in:
  - information,
  - computational power,
  - access to hard instances.

This work

RESEARCH QUESTION

Under what *general conditions* is realizable online learning possible?

This work

RESEARCH QUESTION

Under what *general conditions* is realizable online learning possible?

- How much do we need to relax the worst-case adversary?

Smoothed analysis of algorithms

Spielman and Teng (2004) initiated the analysis of algorithms beyond the worst-case.

Smoothed analysis of algorithms

Spielman and Teng (2004) initiated the analysis of algorithms beyond the worst-case.

- Worst-case analysis:
  - Show that there exists at least one hard problem instance.

Smoothed analysis of algorithms

Spielman and Teng (2004) initiated the analysis of algorithms beyond the worst-case.

- Worst-case analysis:
  - Show that there exists at least one hard problem instance.
  - This can fail to capture the actual behavior of the algorithm in practice.

Smoothed analysis of algorithms

Spielman and Teng (2004) initiated the analysis of algorithms beyond the worst-case.

- Worst-case analysis:
  - Show that there exists at least one hard problem instance.
  - This can fail to capture the actual behavior of the algorithm in practice.
- Non-worst-case analysis:
  - Introduce a (probability) measure over problem instances.

Smoothed analysis of algorithms

Spielman and Teng (2004) initiated the analysis of algorithms beyond the worst-case.

- Worst-case analysis:
  - Show that there exists at least one hard problem instance.
  - This can fail to capture the actual behavior of the algorithm in practice.
- Non-worst-case analysis:
  - Introduce a (probability) measure over problem instances.
  - Show that almost all problems are easy (the hard instances have measure zero).
    - Or, problems are easy with high probability/on average.

Smoothed adversary for online learning

SMOOTHED ADVERSARY LOOP

For $t = 1, 2, \ldots$

Smoothed adversary for online learning

SMOOTHED ADVERSARY LOOP

For $t = 1, 2, \ldots$

- select test distribution μ_t over all instances $\mathcal{X}$

Smoothed adversary for online learning

SMOOTHED ADVERSARY LOOP

For $t = 1, 2, \ldots$

- **select** test distribution μ_t over all instances $\mathcal{X}$
- **draw** test instance $x_t \sim \mu_t$

Smoothed adversary for online learning

SMOOTHED ADVERSARY LOOP

For $t = 1, 2, \ldots$

- **select** test distribution μ_t over all instances $\mathcal{X}$
- **draw** test instance $x_t \sim \mu_t$
- **observe** (fully/partially) learner's internal state after update

Smoothed adversary for online learning

SMOOTHED ADVERSARY LOOP

For $t = 1, 2, \ldots$

- select test distribution μ_t over all instances $\mathcal{X}$
- draw test instance $x_t \sim \mu_t$
- observe (fully/partially) learner's internal state after update

The smoothed online setting is also studied by Rakhlin et al. (2011); Haghtalab et al. (2020).

Smoothed adversary for online learning

- The smoothed adversary framework interpolates between:
  - the i.i.d. setting: μ_t is fixed for all time t

Smoothed adversary for online learning

- The smoothed adversary framework interpolates between:
  - the i.i.d. setting: μ_t is fixed for all time t
  - the worst-case setting: μ_t may be point masses

Example: Gaussian perturbation model

GAUSSIAN-SMOOTHED ADVERSARY:

- adversary selects $\overline{x}$

Example: Gaussian perturbation model

GAUSSIAN-SMOOTHED ADVERSARY:

- adversary selects $\overline{x}$
- test instance x is a perturbed version $\overline{x} + \xi$ where $\xi \sim \mathcal{N}(0, \sigma^2 I)$, so:

$$\mu = \mathcal{N}(\bar{x}, \sigma^2 I).$$

Example: σ-smoothed adversary

σ-SMOOTHED ADVERSARY:

- let ν be an underlying distribution over $\mathcal{X}$

Example: σ-smoothed adversary

σ-SMOOTHED ADVERSARY:

- let ν be an underlying distribution over $\mathcal{X}$
- the adversary can select any distribution μ satisfying:

$$\mu(A) \leq \frac{1}{\sigma} \cdot \nu(A),$$

for all $A \subset \mathcal{X}$ measurable.

Dominated adversary

In this work, we generalize both by the ν-*dominated adversary*.

- **INTUITION:** the dominated adversary cannot place a constant probability mass on an arbitrarily small region of problem instances.

Dominated adversary

In this work, we generalize both by the ν-*dominated adversary*.

- INTUITION: the dominated adversary cannot place a constant probability mass on an arbitrarily small region of problem instances.
- SETTING: let $(\mathcal{X}, \nu)$ be a measure space.

Dominated adversary

In this work, we generalize both by the ν*-dominated adversary*.

- **INTUITION:** the dominated adversary cannot place a constant probability mass on an arbitrarily small region of problem instances.
- **SETTING:** let $(\mathcal{X}, \nu)$ be a measure space.

Definition (Dominated adversary)

The measure ν uniformly dominates a family $\mathcal{M}$ of probability distributions on $\mathcal{X}$ if for all $\varepsilon > 0$ there exists $\delta > 0$ such that:

$$\nu(A) < \delta \quad \Longrightarrow \quad \mu(A) < \varepsilon,$$

for all $A \subset \mathcal{X}$ measurable and distribution $\mu \in \mathcal{M}$.

Dominated adversary

In this work, we generalize both by the *ν-dominated adversary.*

- **INTUITION:** the dominated adversary cannot place a constant probability mass on an arbitrarily small region of problem instances.
- **SETTING:** let $(\mathcal{X}, \nu)$ be a measure space.

Definition (Dominated adversary)

The measure ν uniformly dominates a family $\mathcal{M}$ of probability distributions on $\mathcal{X}$ if for all $\varepsilon > 0$ there exists $\delta > 0$ such that:

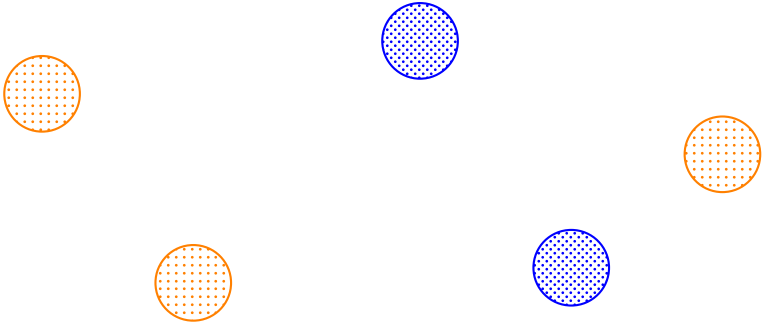
$$\nu(A) < \delta \quad \Longrightarrow \quad \mu(A) < \varepsilon,$$

for all $A \subset \mathcal{X}$ measurable and distribution $\mu \in \mathcal{M}$. We say that adversary is ν-dominated if at all times t it selects μ_t from a family of distributions uniformly dominated by ν.

Example: learning labels for well-separated clusters

SETTING

Suppose that the instance space $\mathcal{X}$ consists of countably many well-separated clusters

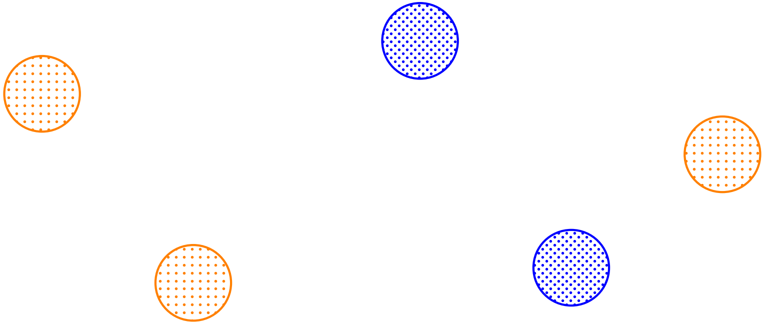


Example: learning labels for well-separated clusters

SETTING

Suppose that the instance space $\mathcal{X}$ consists of countably many well-separated clusters

- within-cluster distances $\ll$ between-cluster distances

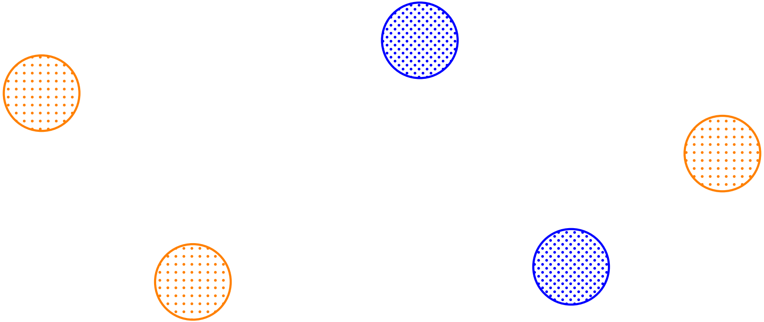


Example: learning labels for well-separated clusters

SETTING

Suppose that the instance space $\mathcal{X}$ consists of countably many well-separated clusters

- within-cluster distances $\ll$ between-cluster distances
- the labels for each cluster is pure (all positive or all negative labels).



Example: learning labels for well-separated clusters

SETTING

Suppose that the instance space $\mathcal{X}$ consists of countably many well-separated clusters

- within-cluster distances $\ll$ between-cluster distances
- the labels for each cluster is pure (all positive or all negative labels).

THE NEAREST NEIGHBOR LEARNER

makes at most one mistake made per cluster

Example: learning labels for well-separated clusters

CONVERGENCE RESULT FOR WELL-SEPARATED CLUSTERS

Let ν be a finite measure on $\mathcal{X}$.

Example: learning labels for well-separated clusters

CONVERGENCE RESULT FOR WELL-SEPARATED CLUSTERS

Let ν be a finite measure on $\mathcal{X}$. The nearest neighbor learner achieves vanishing error rate against any ν-dominated adversary.

Example: learning labels for well-separated clusters

Proof sketch.

- Split $\mathcal{X}$ into two pieces $\mathcal{X}_{\text{easy}} \cup \mathcal{X}_{\text{small}}$, where:

Example: learning labels for well-separated clusters

Proof sketch.

- Split $\mathcal{X}$ into two pieces $\mathcal{X}_{\text{easy}} \cup \mathcal{X}_{\text{small}}$, where:
  - $\mathcal{X}_{\text{easy}}$ is a finite union of clusters

Example: learning labels for well-separated clusters

Proof sketch.

- Split $\mathcal{X}$ into two pieces $\mathcal{X}_{\text{easy}} \cup \mathcal{X}_{\text{small}}$, where:
  - $\mathcal{X}_{\text{easy}}$ is a finite union of clusters
  - $\mathcal{X}_{\text{small}}$ contains very little mass $\nu(\mathcal{X}_{\text{small}}) < \delta$.

Example: learning labels for well-separated clusters

Proof sketch.

- Split $\mathcal{X}$ into two pieces $\mathcal{X}_{\text{easy}} \cup \mathcal{X}_{\text{small}}$, where:
  - $\mathcal{X}_{\text{easy}}$ is a finite union of clusters
  - $\mathcal{X}_{\text{small}}$ contains very little mass $\nu(\mathcal{X}_{\text{small}}) < \delta$.
    - Such a decomposition exists for any $\delta > 0$ by the finiteness of ν.

Example: learning labels for well-separated clusters

Proof sketch.

- Split $\mathcal{X}$ into two pieces $\mathcal{X}_{\text{easy}} \cup \mathcal{X}_{\text{small}}$, where:
  - $\mathcal{X}_{\text{easy}}$ is a finite union of clusters
  - $\mathcal{X}_{\text{small}}$ contains very little mass $\nu(\mathcal{X}_{\text{small}}) < \delta$.
    - Such a decomposition exists for any $\delta > 0$ by the finiteness of ν.
- Nearest neighbor makes finitely many mistakes on $\mathcal{X}_{\text{easy}}$.

Example: learning labels for well-separated clusters

Proof sketch.

- Split $\mathcal{X}$ into two pieces $\mathcal{X}_{\text{easy}} \cup \mathcal{X}_{\text{small}}$, where:
  - $\mathcal{X}_{\text{easy}}$ is a finite union of clusters
  - $\mathcal{X}_{\text{small}}$ contains very little mass $\nu(\mathcal{X}_{\text{small}}) < \delta$.
    - Such a decomposition exists for any $\delta > 0$ by the finiteness of ν.
- Nearest neighbor makes finitely many mistakes on $\mathcal{X}_{\text{easy}}$.
  - These mistakes contribute nothing to the asymptotic mistake rate.

Example: learning labels for well-separated clusters

Proof sketch.

- Split $\mathcal{X}$ into two pieces $\mathcal{X}_{\text{easy}} \cup \mathcal{X}_{\text{small}}$, where:
  - $\mathcal{X}_{\text{easy}}$ is a finite union of clusters
  - $\mathcal{X}_{\text{small}}$ contains very little mass $\nu(\mathcal{X}_{\text{small}}) < \delta$.
    - Such a decomposition exists for any $\delta > 0$ by the finiteness of ν.
- Nearest neighbor makes finitely many mistakes on $\mathcal{X}_{\text{easy}}$.
  - These mistakes contribute nothing to the asymptotic mistake rate.
- The ν-dominated adversary selects points from $\mathcal{X}_{\text{small}}$ at rate $\mu(\mathcal{X}_{\text{small}}) < \varepsilon$.

Example: learning labels for well-separated clusters

Proof sketch.

- Split $\mathcal{X}$ into two pieces $\mathcal{X}_{\mathrm{easy}} \cup \mathcal{X}_{\mathrm{small}}$, where:
  - $\mathcal{X}_{\mathrm{easy}}$ is a finite union of clusters
  - $\mathcal{X}_{\mathrm{small}}$ contains very little mass $\nu(\mathcal{X}_{\mathrm{small}}) < \delta$.
    - Such a decomposition exists for any $\delta > 0$ by the finiteness of ν.
- Nearest neighbor makes finitely many mistakes on $\mathcal{X}_{\mathrm{easy}}$.
  - These mistakes contribute nothing to the asymptotic mistake rate.
- The ν-dominated adversary selects points from $\mathcal{X}_{\mathrm{small}}$ at rate $\mu(\mathcal{X}_{\mathrm{small}}) < \varepsilon$.
  - By the law of large number, at most an ε-fraction of $(x_t)_t$ comes from $\mathcal{X}_{\mathrm{small}}$.

Example: learning labels for well-separated clusters

Proof sketch.

- Split $\mathcal{X}$ into two pieces $\mathcal{X}_{\text{easy}} \cup \mathcal{X}_{\text{small}}$, where:
  - $\mathcal{X}_{\text{easy}}$ is a finite union of clusters
  - $\mathcal{X}_{\text{small}}$ contains very little mass $\nu(\mathcal{X}_{\text{small}}) < \delta$.
    - Such a decomposition exists for any $\delta > 0$ by the finiteness of ν.
- Nearest neighbor makes finitely many mistakes on $\mathcal{X}_{\text{easy}}$.
  - These mistakes contribute nothing to the asymptotic mistake rate.
- The ν-dominated adversary selects points from $\mathcal{X}_{\text{small}}$ at rate $\mu(\mathcal{X}_{\text{small}}) < \varepsilon$.
  - By the law of large number, at most an ε-fraction of $(x_t)_t$ comes from $\mathcal{X}_{\text{small}}$.
- Thus, the asymptotic mistake rate is upper bounded by ε almost surely.

Example: learning labels for well-separated clusters

Proof sketch.

- Split $\mathcal{X}$ into two pieces $\mathcal{X}_{\text{easy}} \cup \mathcal{X}_{\text{small}}$, where:
  - $\mathcal{X}_{\text{easy}}$ is a finite union of clusters
  - $\mathcal{X}_{\text{small}}$ contains very little mass $\nu(\mathcal{X}_{\text{small}}) < \delta$.
    - Such a decomposition exists for any $\delta > 0$ by the finiteness of ν.
- Nearest neighbor makes finitely many mistakes on $\mathcal{X}_{\text{easy}}$.
  - These mistakes contribute nothing to the asymptotic mistake rate.
- The ν-dominated adversary selects points from $\mathcal{X}_{\text{small}}$ at rate $\mu(\mathcal{X}_{\text{small}}) < \varepsilon$.
  - By the law of large number, at most an ε-fraction of $(x_t)_t$ comes from $\mathcal{X}_{\text{small}}$.
- Thus, the asymptotic mistake rate is upper bounded by ε almost surely.
  - Simultaneously apply upper bound for a countable collection of $\varepsilon_k \downarrow 0$.

Example: learning labels for well-separated clusters

Proof sketch.

- Split $\mathcal{X}$ into two pieces $\mathcal{X}_{\text{easy}} \cup \mathcal{X}_{\text{small}}$, where:
  - $\mathcal{X}_{\text{easy}}$ is a finite union of clusters
  - $\mathcal{X}_{\text{small}}$ contains very little mass $\nu(\mathcal{X}_{\text{small}}) < \delta$.
    - Such a decomposition exists for any $\delta > 0$ by the finiteness of ν.
- Nearest neighbor makes finitely many mistakes on $\mathcal{X}_{\text{easy}}$.
  - These mistakes contribute nothing to the asymptotic mistake rate.
- The ν-dominated adversary selects points from $\mathcal{X}_{\text{small}}$ at rate $\mu(\mathcal{X}_{\text{small}}) < \varepsilon$.
  - By the law of large number, at most an ε-fraction of $(x_t)_t$ comes from $\mathcal{X}_{\text{small}}$.
- Thus, the asymptotic mistake rate is upper bounded by ε almost surely.
  - Simultaneously apply upper bound for a countable collection of $\varepsilon_k \downarrow 0$.

The asymptotic mistake rate is zero. $\square$

Generalizing the argument

The argument works even if the clusters are not well-separated.

Generalizing the argument

The argument works even if the clusters are not well-separated.

KEY PROPERTY USED

The nearest neighbor learner makes at most one mistake per cluster.

Generalizing the argument

The argument works even if the clusters are not well-separated.

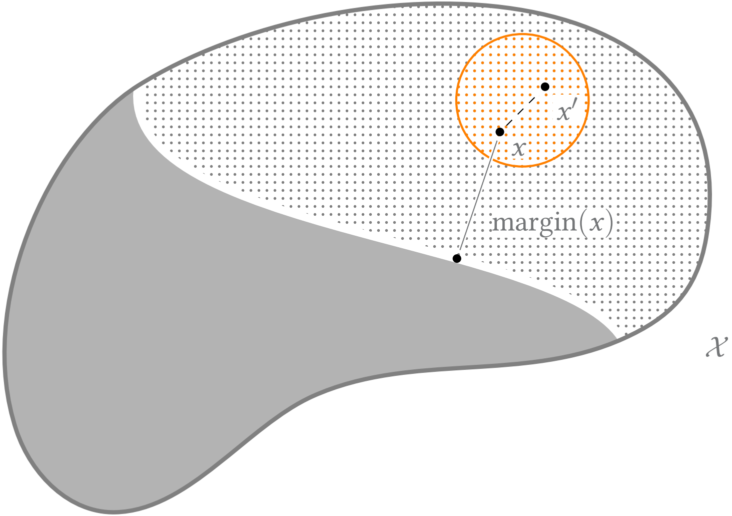
KEY PROPERTY USED

The nearest neighbor learner makes at most one mistake per mutually-labeling set.

- We introduce the device of mutually-labeling sets $U \subset \mathcal{X}$ satisfying the property:

$$\text{interpoint distances in } U < \text{distance to points with different labels.}$$

Mutually-labeling set



Generalizing argument

Definition (Mutually-labeling set)

*A subset $U \subset \mathcal{X}$ is **mutually labeling** if for all $x, x' \in U$:*

$$\underbrace{\rho(x, x')}_{\textit{interpoint distances}} \quad < \quad \underbrace{\mathrm{margin}(x)}_{\textit{distance to decision boundary}}$$

where $\mathrm{margin}(x)$ *is the smallest distance between x and points with different labels:*

$$\mathrm{margin}(x) = \inf\{\rho(x, \overline{x}) : f(x) \neq f(\overline{x})\}.$$

Convergence result

SETTING

Let $(\mathcal{X}, \rho, \nu)$ be a space equipped with a separable metric ρ and a finite Borel measure ν.

Convergence result

SETTING

Let $(\mathcal{X}, \rho, \nu)$ be a space equipped with a separable metric ρ and a finite Borel measure ν.

- Let $f : \mathcal{X} \to \mathcal{Y}$ have decision boundary $\partial\mathcal{X} := \{\text{margin}(x) = 0\}$.

Convergence result

SETTING

Let $(\mathcal{X}, \rho, \nu)$ be a space equipped with a separable metric ρ and a finite Borel measure ν.

- Let $f : \mathcal{X} \to \mathcal{Y}$ have decision boundary $\partial\mathcal{X} := \{\mathrm{margin}(x) = 0\}$.
- Assume $\partial\mathcal{X}$ has ν-measure zero.
  - e.g. The decision boundary is not a space-filling curve.

Convergence result

SETTING

Let $(\mathcal{X}, \rho, \nu)$ be a space equipped with a separable metric ρ and a finite Borel measure ν.

- Let $f : \mathcal{X} \to \mathcal{Y}$ have decision boundary $\partial\mathcal{X} := \{\mathrm{margin}(x) = 0\}$.
- Assume $\partial\mathcal{X}$ has ν-measure zero.
  - e.g. The decision boundary is not a space-filling curve.

Theorem (Convergence of nearest neighbor)

The nearest neighbor rule achieves vanishing mistake rate against a ν-dominated adversary:

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{1}\{\hat{y}_t \neq y_t\} = 0 \quad \text{a.s.}$$

Convergence result

Proof sketch.

- Sufficiently small open balls around non-boundary points are mutually-labeling.

Convergence result

Proof sketch.

- Sufficiently small open balls around non-boundary points are mutually-labeling.
- There is an a.e.-countable cover of $\mathcal{X}$ by these balls:

Convergence result

Proof sketch.

- Sufficiently small open balls around non-boundary points are mutually-labeling.
- There is an a.e.-countable cover of $\mathcal{X}$ by these balls:
  - Use separability of $\mathcal{X}$ and that $\partial\mathcal{X}$ has measure zero.

Convergence result

Proof sketch.

- Sufficiently small open balls around non-boundary points are mutually-labeling.
- There is an a.e.-countable cover of $\mathcal{X}$ by these balls:
  - Use separability of $\mathcal{X}$ and that $\partial\mathcal{X}$ has measure zero.
- Decompose $\mathcal{X}$ into $\mathcal{X}_{\text{easy}} \cup \mathcal{X}_{\text{small}}$ like before:

Convergence result

Proof sketch.

- Sufficiently small open balls around non-boundary points are mutually-labeling.
- There is an a.e.-countable cover of $\mathcal{X}$ by these balls:
  - Use separability of $\mathcal{X}$ and that $\partial\mathcal{X}$ has measure zero.
- Decompose $\mathcal{X}$ into $\mathcal{X}_{\text{easy}} \cup \mathcal{X}_{\text{small}}$ like before:
  - $\mathcal{X}_{\text{easy}}$ is a finite union of these balls.

Convergence result

Proof sketch.

- Sufficiently small open balls around non-boundary points are mutually-labeling.
- There is an a.e.-countable cover of $\mathcal{X}$ by these balls:
  - Use separability of $\mathcal{X}$ and that $\partial\mathcal{X}$ has measure zero.
- Decompose $\mathcal{X}$ into $\mathcal{X}_{\text{easy}} \cup \mathcal{X}_{\text{small}}$ like before:
  - $\mathcal{X}_{\text{easy}}$ is a finite union of these balls.
  - $\mathcal{X}_{\text{small}}$ contains very little mass $\nu(\mathcal{X}_{\text{small}}) < \delta$.

Convergence result

Proof sketch.

- Sufficiently small open balls around non-boundary points are mutually-labeling.
- There is an a.e.-countable cover of $\mathcal{X}$ by these balls:
  - Use separability of $\mathcal{X}$ and that $\partial\mathcal{X}$ has measure zero.
- Decompose $\mathcal{X}$ into $\mathcal{X}_{\text{easy}} \cup \mathcal{X}_{\text{small}}$ like before:
  - $\mathcal{X}_{\text{easy}}$ is a finite union of these balls.
  - $\mathcal{X}_{\text{small}}$ contains very little mass $\nu(\mathcal{X}_{\text{small}}) < \delta$.

By prior argument, the mistake rate converges to zero almost surely. $\square$

Discussion about this work

UPSHOT

1. The nearest neighbor rule works fine against the ν-dominated adversary.

Discussion about this work

UPSHOT

1. The nearest neighbor rule works fine against the ν-dominated adversary.
   - Along the way, we obtained a nice analytic tool (mutually-labeling sets).

Discussion about this work

UPSHOT

1. The nearest neighbor rule works fine against the ν-dominated adversary.
   - Along the way, we obtained a nice analytic tool (mutually-labeling sets).

2. The argument generalizes to other certain types of online learners (see paper).

Discussion about this work

UPSHOT

1. The nearest neighbor rule works fine against the ν-dominated adversary.
   - Along the way, we obtained a nice analytic tool (mutually-labeling sets).
2. The argument generalizes to other certain types of online learners (see paper).
   - We give a sufficient condition to online learning against a dominated adversary.

Discussion about this work

UPSHOT

1. The nearest neighbor rule works fine against the ν-dominated adversary.
   - Along the way, we obtained a nice analytic tool (mutually-labeling sets).

2. The argument generalizes to other certain types of online learners (see paper).
   - We give a sufficient condition to online learning against a dominated adversary.

3. It is easy to convert asymptotic result to a rate of convergence (see paper).

Discussion about this work

UPSHOT

1. The nearest neighbor rule works fine against the ν-dominated adversary.
   - Along the way, we obtained a nice analytic tool (mutually-labeling sets).
2. The argument generalizes to other certain types of online learners (see paper).
   - We give a sufficient condition to online learning against a dominated adversary.
3. It is easy to convert asymptotic result to a rate of convergence (see paper).
   - Quantify geometry of the the instance space and concept to be learned
     - doubling dimension of space and Minkowski content of the boundary

Discussion about this work

UPSHOT

1. The nearest neighbor rule works fine against the ν-dominated adversary.
   - Along the way, we obtained a nice analytic tool (mutually-labeling sets).

2. The argument generalizes to other certain types of online learners (see paper).
   - We give a sufficient condition to online learning against a dominated adversary.

3. It is easy to convert asymptotic result to a rate of convergence (see paper).
   - Quantify geometry of the the instance space and concept to be learned
     - doubling dimension of space and Minkowski content of the boundary
   - Quantify the strength of the adversary
     - Smoothness rate in definition of a dominated adversary $\varepsilon(\delta)$

Further work

Open questions

QUESTIONS

- Does the ν-dominated adversary balance between generality and tractability well?

Open questions

QUESTIONS

- Does the ν-dominated adversary balance between generality and tractability well?
- Is smoothed online learning possible when there is benign label noise?

Online learning with noise

ONLINE LEARNING LOOP

For $t = 1, 2, \ldots$

- receive instance x_t
- predict label $\hat{y}_t$
- observe label $y_t \sim P_{Y|X=x_t}$ drawn from a fixed conditional distribution
- incur loss $\ell(x_t, y_t, \hat{y}_t)$

Online learning with noise

SETTING

Let data $(x_1, y_1), \ldots, (x_t, y_t)$ be adaptively generated:

Online learning with noise

SETTING

Let data $(x_1, y_1), \ldots, (x_t, y_t)$ be adaptively generated:

- x_t can depend arbitrarily on the past $\{(x_\tau, y_\tau)\}_{\tau=1}^{t-1}$

Online learning with noise

SETTING

Let data $(x_1, y_1), \ldots, (x_t, y_t)$ be adaptively generated:

- x_t can depend arbitrarily on the past $\{(x_\tau, y_\tau)\}_{\tau=1}^{t-1}$
- $y_t \sim P_{Y|X=x_t}$ is conditionally independent given x_t.

Online learning with noise

SETTING

Let data $(x_1, y_1), \ldots, (x_t, y_t)$ be adaptively generated:

- x_t can depend arbitrarily on the past $\{(x_\tau, y_\tau)\}_{\tau=1}^{t-1}$
- $y_t \sim P_{Y|X=x_t}$ is conditionally independent given x_t.

QUESTION

How should this data be used to construct a classifier?

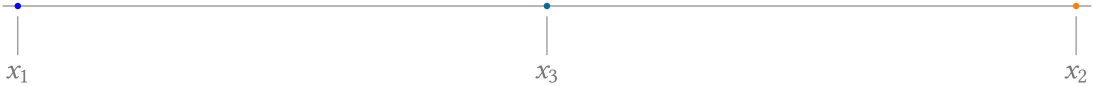
Online learning with noise: binary search on noise

BINARY SEARCH SAMPLING ALGORITHM

Online learning with noise: binary search on noise

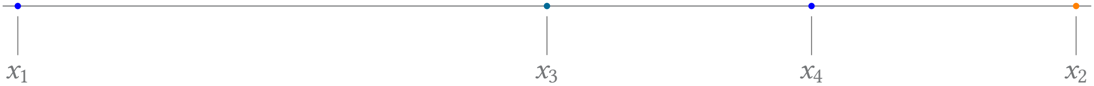
BINARY SEARCH SAMPLING ALGORITHM

Online learning with noise: binary search on noise



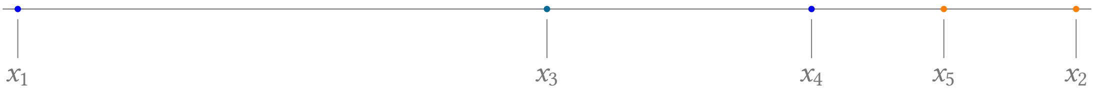
BINARY SEARCH SAMPLING ALGORITHM

Online learning with noise: binary search on noise



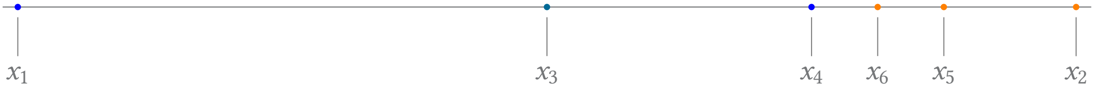
BINARY SEARCH SAMPLING ALGORITHM

Online learning with noise: binary search on noise



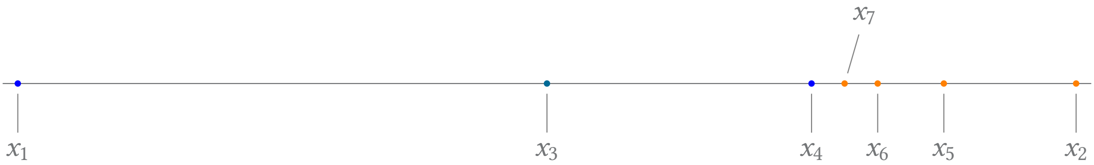
BINARY SEARCH SAMPLING ALGORITHM

Online learning with noise: binary search on noise



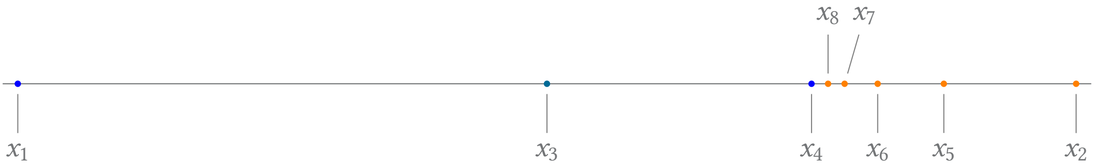
BINARY SEARCH SAMPLING ALGORITHM

Online learning with noise: binary search on noise



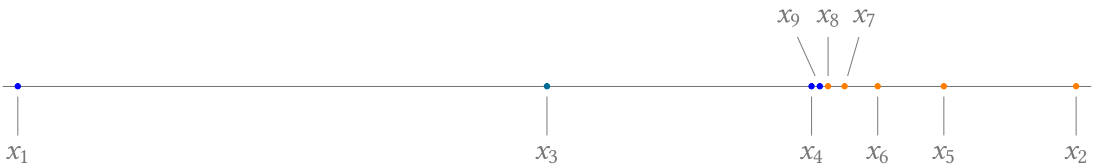
BINARY SEARCH SAMPLING ALGORITHM

Online learning with noise: binary search on noise



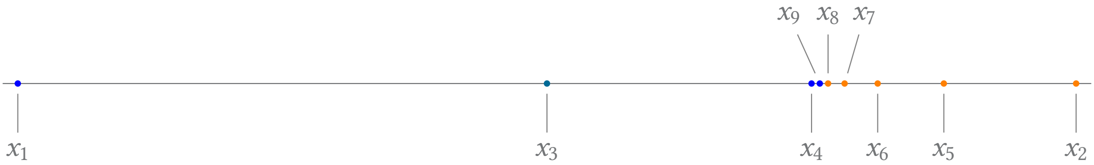
BINARY SEARCH SAMPLING ALGORITHM

Online learning with noise: binary search on noise



BINARY SEARCH SAMPLING ALGORITHM

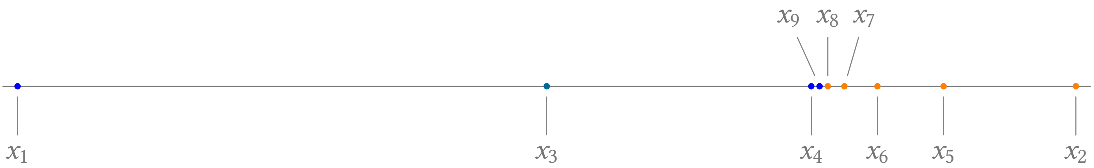
Online learning with noise: binary search on noise



BINARY SEARCH SAMPLING ALGORITHM

For $t = 1, 2, \ldots$

Online learning with noise: binary search on noise

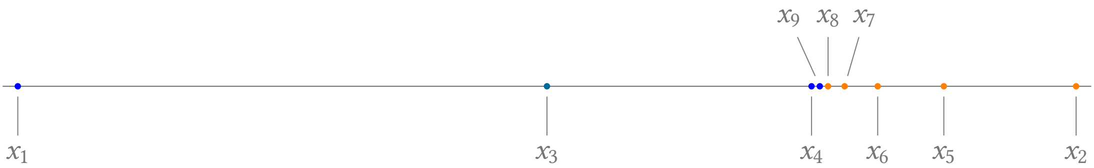


BINARY SEARCH SAMPLING ALGORITHM

For $t = 1, 2, \ldots$

- $x_- \leftarrow$ max negative data point in data set

Online learning with noise: binary search on noise

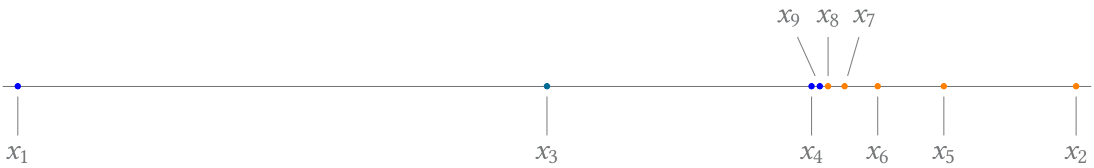


BINARY SEARCH SAMPLING ALGORITHM

For $t = 1, 2, \ldots$

- $x_- \leftarrow$ max negative data point in data set
- $x_+ \leftarrow$ min positive data point in data set

Online learning with noise: binary search on noise



BINARY SEARCH SAMPLING ALGORITHM

For $t = 1, 2, \ldots$

- $x_- \leftarrow$ max negative data point in data set
- $x_+ \leftarrow$ min positive data point in data set
- $x_{t+1} \leftarrow \text{mean}(x_-, x_+)$

Online learning with noise: binary search on noise

TWO INDISTINGUISHABLE WORLDS

An unbounded adversary can select points in a way so that the learner can't distinguish:

Online learning with noise: binary search on noise

TWO INDISTINGUISHABLE WORLDS

An unbounded adversary can select points in a way so that the learner can't distinguish:

1. the labels are generated by a threshold function $f(x) = \mathbb{1}\{x \geq \theta\}$

Online learning with noise: binary search on noise

TWO INDISTINGUISHABLE WORLDS

An unbounded adversary can select points in a way so that the learner can't distinguish:

1. the labels are generated by a threshold function $f(x) = \mathbb{1}\{x \geq \theta\}$
2. the labels are drawn from $\mathrm{Ber}(\frac{1}{2} + \Delta)$

Online learning with noise: binary search on noise

Suppose labels are drawn from pure noise $\mathrm{Ber}(\frac{1}{2})$.

Online learning with noise: binary search on noise

Suppose labels are drawn from pure noise $\mathrm{Ber}(\frac{1}{2})$.

I.I.D. SETTING: for all time t, all intervals I simultaneously satisfy with high probability:

Online learning with noise: binary search on noise

Suppose labels are drawn from pure noise $\mathrm{Ber}(\frac{1}{2})$.

i.i.d. setting: for all time t, all intervals I simultaneously satisfy with high probability:

$$\frac{\#\text{ positive data points in } I}{\#\text{ data points in } I} = \frac{1}{2} \pm O\left(\sqrt{\frac{\log t}{\#\text{ data points in } I}}\right).$$

Online learning with noise: binary search on noise

Suppose labels are drawn from pure noise $\mathrm{Ber}(\frac{1}{2})$.

I.I.D. SETTING: for all time t, all intervals I simultaneously satisfy with high probability:

$$\frac{\#\text{ positive data points in } I}{\#\text{ data points in } I} = \frac{1}{2} \pm O\left(\sqrt{\frac{\log t}{\#\text{ data points in } I}}\right).$$

- The class of all intervals has VC dimension 2.

Online learning with noise: binary search on noise

Suppose labels are drawn from pure noise $\mathrm{Ber}(\frac{1}{2})$.

I.I.D. SETTING: for all time t, all intervals I simultaneously satisfy with high probability:

$$\frac{\#\text{ positive data points in } I}{\#\text{ data points in } I} = \frac{1}{2} \pm O\left(\sqrt{\frac{\log t}{\#\text{ data points in } I}}\right).$$

- The class of all intervals has VC dimension 2.

SEQUENTIAL SETTING: at each time t, the interval $I = [0, x_t]$ satisfies:

Online learning with noise: binary search on noise

Suppose labels are drawn from pure noise $\mathrm{Ber}(\frac{1}{2})$.

I.I.D. SETTING: for all time t, all intervals I simultaneously satisfy with high probability:

$$\frac{\#\text{ positive data points in } I}{\#\text{ data points in } I} = \frac{1}{2} \pm O\left(\sqrt{\frac{\log t}{\#\text{ data points in } I}}\right).$$

- The class of all intervals has VC dimension 2.

SEQUENTIAL SETTING: at each time t, the interval $I = [0, x_t]$ satisfies:

$$\frac{\#\text{ positive data points in } I}{\#\text{ data points in } I} = 0,$$

where $\#\text{ data points in } I \approx \frac{1}{2}t$.

Online learning with noise: binary search on noise

Suppose labels are drawn from pure noise $\mathrm{Ber}(\frac{1}{2})$.

I.I.D. SETTING: for all time t, all intervals I simultaneously satisfy with high probability:

$$\frac{\#\text{ positive data points in } I}{\#\text{ data points in } I} = \frac{1}{2} \pm O\left(\sqrt{\frac{\log t}{\#\text{ data points in } I}}\right).$$

- The class of all intervals has VC dimension 2.

SEQUENTIAL SETTING: at each time t, the interval $I = [0, x_t]$ satisfies:

$$\frac{\#\text{ positive data points in } I}{\#\text{ data points in } I} = 0,$$

where $\#$ data points in $I \approx \frac{1}{2}t$.

- For a vast majority of intervals with $< \frac{1}{2}t$ points, the average label is far from $\frac{1}{2}$.

Challenge of online learning with noise

MAKING PATTERNS OUT OF NOISE

In the sequential setting, the uniform law of large number can fail

Challenge of online learning with noise

MAKING PATTERNS OUT OF NOISE

In the sequential setting, the uniform law of large number can fail

- there can be many balls/intervals whose average label is far from correct

Challenge of online learning with noise

MAKING PATTERNS OUT OF NOISE

In the sequential setting, the uniform law of large number can fail

- there can be many balls/intervals whose average label is far from correct
- finite VC dimension does not imply sequential uniform Glivenko-Cantelli property

The k_n-nearest neighbor rule

K_N-NEAREST NEIGHBOR ALGORITHM

The k_n-nearest neighbor rule

k_n-NEAREST NEIGHBOR ALGORITHM

- remember all past data points $\{(x_1, y_1), \ldots, (x_n, y_n)\}$

The k_n-nearest neighbor rule

k_N-NEAREST NEIGHBOR ALGORITHM

- remember all past data points $\{(x_1, y_1), \ldots, (x_n, y_n)\}$
- given query x, find the k_n most similar data point in memory

The k_n-nearest neighbor rule

k_N-NEAREST NEIGHBOR ALGORITHM

- remember all past data points $\{(x_1, y_1), \ldots, (x_n, y_n)\}$
- given query x, find the k_n most similar data point in memory
- predict using majority vote over k_n nearest neighbors

Online learning with noise

SETTING

Let $\mathcal{X} = [0, 1]$ be the unit interval and let $P_{Y|X=x} = \mathrm{Ber}(\frac{1}{2} + \Delta)$.

Online learning with noise

SETTING

Let $\mathcal{X} = [0,1]$ be the unit interval and let $P_{Y|X=x} = \mathrm{Ber}(\frac{1}{2} + \Delta)$.

- Assume that $\Delta \in (0, \frac{1}{2})$ is bounded away from $\frac{1}{2}$.

Online learning with noise

SETTING

Let $\mathcal{X} = [0, 1]$ be the unit interval and let $P_{Y|X=x} = \mathrm{Ber}(\frac{1}{2} + \Delta)$.

- Assume that $\Delta \in (0, \frac{1}{2})$ is bounded away from $\frac{1}{2}$.

Proposition

Suppose that $k_n = o(n)$. An unbounded adversary can adaptively generate a sequence $(x_n)_n$ such that the mistake rate of the k_n-nearest neighbor rule never converges to zero:

Online learning with noise

SETTING

Let $\mathcal{X} = [0, 1]$ be the unit interval and let $P_{Y|X=x} = \mathrm{Ber}(\frac{1}{2} + \Delta)$.

- Assume that $\Delta \in (0, \frac{1}{2})$ is bounded away from $\frac{1}{2}$.

Proposition

Suppose that $k_n = o(n)$. An unbounded adversary can adaptively generate a sequence $(x_n)_n$ such that the mistake rate of the k_n-nearest neighbor rule never converges to zero:

$$\liminf_{n\to\infty} \frac{1}{n} \sum_{j=1}^{n} \mathbb{1}\{\hat{y}_{k_n\text{-}NN}(x_n) \neq 1\} = \Omega(1).$$

Smoothed online learning with noise

QUESTION

How does the k_n-nearest neighbor rule perform against a dominated adversary?

Preliminary result

SETTING

Let $\mathcal{X} = [0, 1]$ be the unit interval and let $P_{Y|X=x} = \mathrm{Ber}(\frac{1}{2} + \Delta)$.

Preliminary result

SETTING

Let $\mathcal{X} = [0,1]$ be the unit interval and let $P_{Y|X=x} = \mathrm{Ber}(\frac{1}{2} + \Delta)$.

- Let the dominated adversary μ be smoothed at rate $\varepsilon : [0,1] \to [0,1]$ so that:

$$\nu(A) < \delta \qquad \Longrightarrow \qquad \mu(A) < \varepsilon(\delta).$$

  Here, ν is the Lebesgue measure on $\mathcal{X}$.

Preliminary result

SETTING

Let $\mathcal{X} = [0,1]$ be the unit interval and let $P_{Y|X=x} = \mathrm{Ber}(\frac{1}{2} + \Delta)$.

- Let the dominated adversary μ be smoothed at rate $\varepsilon : [0,1] \to [0,1]$ so that:

$$\nu(A) < \delta \qquad \Longrightarrow \qquad \mu(A) < \varepsilon(\delta).$$

  Here, ν is the Lebesgue measure on $\mathcal{X}$.

Theorem

Let $k_n = \omega\big((\log \varepsilon^{-1}(\frac{1}{n}))^2\big)$. The k_n-nearest neighbor rule eventually makes no mistakes:

Preliminary result

SETTING

Let $\mathcal{X} = [0,1]$ be the unit interval and let $P_{Y|X=x} = \mathrm{Ber}(\frac{1}{2} + \Delta)$.

- Let the dominated adversary μ be smoothed at rate $\varepsilon : [0,1] \to [0,1]$ so that:

$$\nu(A) < \delta \qquad \Longrightarrow \qquad \mu(A) < \varepsilon(\delta).$$

  Here, ν is the Lebesgue measure on $\mathcal{X}$.

Theorem

Let $k_n = \omega\big((\log \varepsilon^{-1}(\frac{1}{n}))^2\big)$. The k_n-nearest neighbor rule eventually makes no mistakes:

$$\lim_{n\to\infty} \mathbb{1}\{\hat{y}_{k_n\text{-}NN}(x_n) \neq 1\} = 0 \qquad a.s.$$

Proof idea

1. Define the finite family of simple intervals of depth $\ell_n \in \mathbb{N}$ of the form:

$$\left[j_0 2^{-\ell_n}, j_1 2^{-\ell_n}\right], \qquad j_0, j_1 \in \mathbb{N}$$

Proof idea

1. Define the finite family of simple intervals of depth $\ell_n \in \mathbb{N}$ of the form:

$$\left[j_0 2^{-\ell_n}, j_1 2^{-\ell_n}\right], \qquad j_0, j_1 \in \mathbb{N}$$

   - this class satisfies uniform law of large numbers because it is finite

Proof idea

1. Define the finite family of simple intervals of depth $\ell_n \in \mathbb{N}$ of the form:

$$\left[j_0 2^{-\ell_n}, j_1 2^{-\ell_n}\right], \qquad j_0, j_1 \in \mathbb{N}$$

   - this class satisfies uniform law of large numbers because it is finite

2. Approximate any k_n-nearest neighbor interval $I_{k_n\text{-NN}}$ by simple intervals:

$$I_{\text{inner}} \subset I_{k_n\text{-NN}} \subset I_{\text{outer}}$$

Proof idea

1. Define the finite family of simple intervals of depth $\ell_n \in \mathbb{N}$ of the form:

$$\left[j_0 2^{-\ell_n}, j_1 2^{-\ell_n}\right], \qquad j_0, j_1 \in \mathbb{N}$$

   - this class satisfies uniform law of large numbers because it is finite

2. Approximate any k_n-nearest neighbor interval $I_{k_n\text{-NN}}$ by simple intervals:

$$I_{\text{inner}} \subset I_{k_n\text{-NN}} \subset I_{\text{outer}}$$

   - They can be chosen so that $I_{\text{outer}} \setminus I_{\text{inner}}$ is a union of two dyadic intervals of length $2^{-\ell_n}$

Proof idea

1. Define the finite family of simple intervals of depth $\ell_n \in \mathbb{N}$ of the form:

$$\left[j_0 2^{-\ell_n}, j_1 2^{-\ell_n}\right], \qquad j_0, j_1 \in \mathbb{N}$$

   - this class satisfies uniform law of large numbers because it is finite

2. Approximate any k_n-nearest neighbor interval $I_{k_n\text{-NN}}$ by simple intervals:

$$I_{\text{inner}} \subset I_{k_n\text{-NN}} \subset I_{\text{outer}}$$

   - They can be chosen so that $I_{\text{outer}} \setminus I_{\text{inner}}$ is a union of two dyadic intervals of length $2^{-\ell_n}$

3. I_{outer} contains at least k_n points: most are correctly labeled points (uniform LLN)

Proof idea

1. Define the finite family of simple intervals of depth $\ell_n \in \mathbb{N}$ of the form:

$$\left[j_0 2^{-\ell_n}, j_1 2^{-\ell_n}\right], \qquad j_0, j_1 \in \mathbb{N}$$

   - this class satisfies uniform law of large numbers because it is finite

2. Approximate any k_n-nearest neighbor interval $I_{k_n\text{-NN}}$ by simple intervals:

$$I_{\text{inner}} \subset I_{k_n\text{-NN}} \subset I_{\text{outer}}$$

   - They can be chosen so that $I_{\text{outer}} \setminus I_{\text{inner}}$ is a union of two dyadic intervals of length $2^{-\ell_n}$

3. I_{outer} contains at least k_n points: most are correctly labeled points (uniform LLN)
4. I_{inner} differ by very few points (smoothness + union bound over dyadic intervals)

Proof idea

1. Define the finite family of simple intervals of depth $\ell_n \in \mathbb{N}$ of the form:

$$\left[j_0 2^{-\ell_n}, j_1 2^{-\ell_n}\right], \qquad j_0, j_1 \in \mathbb{N}$$

   - this class satisfies uniform law of large numbers because it is finite

2. Approximate any k_n-nearest neighbor interval $I_{k_n\text{-NN}}$ by simple intervals:

$$I_{\text{inner}} \subset I_{k_n\text{-NN}} \subset I_{\text{outer}}$$

   - They can be chosen so that $I_{\text{outer}} \setminus I_{\text{inner}}$ is a union of two dyadic intervals of length $2^{-\ell_n}$

3. I_{outer} contains at least k_n points: most are correctly labeled points (uniform LLN)
4. I_{inner} differ by very few points (smoothness + union bound over dyadic intervals)
5. The majority of points in $I_{k_n\text{-NN}}$ are correctly labeled

Proof idea

1. Define the finite family of simple intervals of depth $\ell_n \in \mathbb{N}$ of the form:

$$\left[j_0 2^{-\ell_n}, j_1 2^{-\ell_n}\right], \qquad j_0, j_1 \in \mathbb{N}$$

   - this class satisfies uniform law of large numbers because it is finite

2. Approximate any k_n-nearest neighbor interval $I_{k_n\text{-NN}}$ by simple intervals:

$$I_{\text{inner}} \subset I_{k_n\text{-NN}} \subset I_{\text{outer}}$$

   - They can be chosen so that $I_{\text{outer}} \setminus I_{\text{inner}}$ is a union of two dyadic intervals of length $2^{-\ell_n}$

3. I_{outer} contains at least k_n points: most are correctly labeled points (uniform LLN)
4. I_{inner} differ by very few points (smoothness + union bound over dyadic intervals)
5. The majority of points in $I_{k_n\text{-NN}}$ are correctly labeled

Our choice of k_n leads to $\Pr(\mathbb{1}\{\text{mistake}_n\}) = o(n^{-1})$. Apply Borel-Cantelli. $\square$

Big picture: smoothed analysis of online learning

Many applications of machine learning happen in the online setting:

Big picture: smoothed analysis of online learning

Many applications of machine learning happen in the online setting:

- never-ending and non-i.i.d. stream of tasks

Big picture: smoothed analysis of online learning

Many applications of machine learning happen in the online setting:

- never-ending and non-i.i.d. stream of tasks
- models are updated incrementally

Big picture: smoothed analysis of online learning

Many applications of machine learning happen in the online setting:

- never-ending and non-i.i.d. stream of tasks
- models are updated incrementally

OPPORTUNITY: we might not live in the worst-case adversarial setting

Big picture: smoothed analysis of online learning

Many applications of machine learning happen in the online setting:

- never-ending and non-i.i.d. stream of tasks
- models are updated incrementally

OPPORTUNITY: we might not live in the worst-case adversarial setting

- Is the ν-dominated online learning setting realistic and tractable?

Big picture: smoothed analysis of online learning

Many applications of machine learning happen in the online setting:

- never-ending and non-i.i.d. stream of tasks
- models are updated incrementally

OPPORTUNITY: we might not live in the worst-case adversarial setting

- Is the ν-dominated online learning setting realistic and tractable?
- If so, can we design and analyze algorithms specifically for this setting?
  - e.g. a minimax optimal algorithm might not be optimal in this setting

Thank you

ACKNOWLEDGEMENTS

Joint work with Sanjoy Dasgupta and Robi Bhattacharjee.

Paper is available at *https://arxiv.org/abs/2307.01170*.

Additional slides

Related work: realizable online learning

LEARNABILITY OF A CONCEPT CLASS

Let $\mathcal{F}$ be a concept class. When is it learnable under worst-case online setting?

- Littlestone (1988): if $\mathcal{F}$ has finite Littlestone dimension d, it is possible to make at most d mistakes (uniform bound over all $f \in \mathcal{F}$)
- Bousquet et al. (2021): if $\mathcal{F}$ does not have an infinite Littlestone tree, it is possible to make finitely many mistakes (no uniform-bound over $f \in \mathcal{F}$)

NON-PARAMETRIC ONLINE LEARNING

Non-parametric classes have infinite Littlestone trees. Any deterministic learner makes a mistake every round in the worst-case.

- We show that online learning is possible under mild smoothing of adversary.
  - Finite Littlestone dimension not needed!

Related work: uniform convergence

I.I.D. UNIFORM CONVERGENCE

- Balsubramani et al. (2019): uniform convergence for empirical conditional measures
  - Let $\mathcal{A}, \mathcal{B} \subset 2^{\mathcal{X}}$ have VC dimensions at most d. At time n, for all $A \in \mathcal{A}$ and $B \in \mathcal{B}$:

$$\left|\hat{\mu}_n(A|B) - \mu(A|B)\right| < O\left(\sqrt{\frac{d\log(n)}{\#\text{ data points in } B}}\right) \quad \text{w.h.p.}$$

SEQUENTIAL UNIFORM CONVERGENCE

- Rakhlin et al. (2015): finite VC dimension is not sufficient for sequential uniform convergence; finite Littlestone dimension necessary and sufficient.
  - Let $(X_n)_n$ be an $(\mathcal{F}_n)_n$-stochastic process and μ_n the conditional law of X_n given $\mathcal{F}_{n-1}$.

$$\forall \varepsilon > 0, \quad \lim_{N\to\infty} \sup_{\mu} \Pr\left(\sup_{n>N} \sup_{A\in\mathcal{A}} \left|\hat{\mu}_n(A) - \frac{1}{n}\sum_{k=1}^{n} \mu_k(A)\right| > \varepsilon\right) = 0$$

Open questions: sequential uniform convergence

1. Sequential uniform convergence for (adaptive) sequences $(\mathcal{A}_n)_n$ of classes $A_n \subset 2^{\mathcal{X}}$?

$$\forall \varepsilon > 0, \quad \lim_{N\to\infty} \sup_{\mu} \Pr\left(\sup_{n>N} \boxed{\sup_{A\in\mathcal{A}_n}} \left| \hat{\mu}_n(A) - \frac{1}{n}\sum_{k=1}^{n} \mu_k(A) \right| > \varepsilon \right) = 0$$

2. Sequential uniform convergence for smoothed processes?
   - Suppose $\mathcal{A}$ is well-approximated by some class $\mathcal{B}$ with finite Littlestone dimension:

$$\sup_{\mathcal{A}} \inf_{\mathcal{B}} \nu\big(B_{\text{outer}}(A) \setminus B_{\text{inner}}(A)\big) < \delta.$$

   Can smoothness extend uniform convergence for $\mathcal{B}$ to $\mathcal{A}$? Does $\mathcal{B}$ need to be closed under set operations, as with the dyadic cubes?

Related work: smoothed online learning

EXISTING RESULTS

- Haghtalab et al. (2022) and Block et al. (2022) show that in the smoothed online setting where the adversary also controls labels, finite VC dimension is sufficient
  - Assumes $\frac{1}{\sigma}$-Lipschitz smoothing: $\mu(A) < \frac{1}{\sigma} \cdot \nu(A)$ for all $A \subset \mathcal{X}$ measurable.
  - Requires knowledge of underlying base measure ν.

OUR RESULT

- Generalizes the Lipschitz adversary to dominated adversary.
- Does not require finite VC/Littlestone dimension.
- Does not need knowledge of base measure ν.
- But, labels are not chosen adaptively (chosen adversarially at beginning of time).

References

Akshay Balsubramani, Sanjoy Dasgupta, Shay Moran, et al. An adaptive nearest neighbor rule for classification. *Advances in Neural Information Processing Systems*, 32, 2019.

Adam Block, Yuval Dagan, Noah Golowich, and Alexander Rakhlin. Smoothed online learning is as easy as statistical learning. In *Conference on Learning Theory*, pages 1716–1786. PMLR, 2022.

Olivier Bousquet, Steve Hanneke, Shay Moran, Ramon van Handel, and Amir Yehudayoff. A theory of universal learning. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 532–541, 2021.

Nika Haghtalab, Tim Roughgarden, and Abhishek Shetty. Smoothed analysis of online and differentially private learning. *Advances in Neural Information Processing Systems*, 33:9203–9215, 2020.

Nika Haghtalab, Tim Roughgarden, and Abhishek Shetty. Smoothed analysis with adaptive adversaries. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 942–953. IEEE, 2022.

Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine learning*, 2(4):285–318, 1988.

Alexander Rakhlin, Karthik Sridharan, and Ambuj Tewari. Online learning: Stochastic and constrained adversaries. *arXiv preprint arXiv:1104.5070*, 2011.

Alexander Rakhlin, Karthik Sridharan, and Ambuj Tewari. Sequential complexities and uniform martingale laws of large numbers. *Probability theory and related fields*, 161:111–153, 2015.

Daniel A Spielman and Shang-Hua Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *Journal of the ACM (JACM)*, 51(3):385–463, 2004.