Algorithmic stability for regression & classification

Rina Foygel Barber (joint with Jake Soloff & Rebecca Willett)

http://rinafb.github.io/

Collaborators

Jake Soloff

Rebecca Willett

Background on algorithmic stability

Background: algorithmic stability

Supervised learning setting

Data $(X_1, Y_1), \ldots, (X_n, Y_n) \in \mathbb{R}^d \times \mathbb{R}$ $\xrightarrow{\text{algorithm } \mathcal{A}}$ Fitted model $\widehat{f}$

Classical results are often of the form:

$$\underbrace{\text{strong assumptions}}_{\substack{\text{e.g., parametric model /} \\ \text{smoothness assumptions / etc}}} \implies \underbrace{\text{strong guarantees}}_{\substack{\text{e.g., } \widehat{f} \approx \text{true model,} \\ \text{with high probability}}}$$

Background: algorithmic stability

More recently, assumption-lean / distribution-free results:

$$\underbrace{\text{weaker assumptions}}_{\substack{\text{e.g., i.i.d. data from} \\ \text{an arbitrary distribution}}} \implies \underbrace{\text{weaker guarantees}}_{\substack{\text{e.g., can provide a prediction} \\ \text{interval around } \widehat{f}}}$$

Background: algorithmic stability

More recently, assumption-lean / distribution-free results:

$$\underbrace{\text{weaker assumptions}}_{\substack{\text{e.g., i.i.d. data from} \\ \text{an arbitrary distribution}}} \implies \underbrace{\text{weaker guarantees}}_{\substack{\text{e.g., can provide a prediction} \\ \text{interval around } \widehat{f}}}$$

- Generally do not guarantee or require concentration / consistency:

$$\widehat{f} \approx \widehat{f}' \text{ if we resample entire data set}$$

- But, often need a milder condition—stability:

$$\widehat{f} \approx \widehat{f}' \text{ if we resample small fraction of data set}$$

Background: algorithmic stability

Definition: algorithmic stability

$\mathcal{A}$ is (ϵ, δ)-stable if, for any dataset $\mathcal{D}$ of size n and any x,

$$\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\left\{\left|\widehat{f}(x) - \widehat{f}_{-i}(x)\right| \geq \epsilon\right\} \leq \delta$$

where $\widehat{f} = \mathcal{A}(\mathcal{D})$ and $\widehat{f}_{-i} = \mathcal{A}(\mathcal{D}_{-i})$

Intuition: only a small fraction of data points can be highly influential

Background: algorithmic stability

Definition: algorithmic stability

$\mathcal{A}$ is (ϵ, δ)-stable if, for any dataset $\mathcal{D}$ of size n and any x,

$$\frac{1}{n}\sum_{i=1}^{n} \mathbb{P}_{\xi}\left\{\left|\widehat{f}(x) - \widehat{f}_{-i}(x)\right| \geq \epsilon\right\} \leq \delta$$

where $\widehat{f} = \mathcal{A}(\mathcal{D}; \xi)$ and $\widehat{f}_{-i} = \mathcal{A}(\mathcal{D}_{-i}; \xi)$

random seed $\xi \sim$ Unif[0, 1]

Intuition: only a small fraction of data points can be highly influential

Background: algorithmic stability

For statistical settings, a weaker definition is sufficient:

Definition: algorithmic stability with respect to P

For $\mathcal{D} \sim P^n$ and an independent test point $X \sim P_X$,

$$\frac{1}{n}\sum_{i=1}^{n} \mathbb{P}\left\{\left|\widehat{f}(X) - \widehat{f}_{-i}(X)\right| \geq \epsilon\right\} \leq \delta$$

[Background: Kearns & Ron 1999; Bousquet & Elisseeff 2002; Elisseeff et al. 2005]

Background: algorithmic stability

Stability has implications for:

- Generalization [Bousquet & Elisseeff 2002; Elisseeff et al. 2005]
- Learnability [Shalev-Shwartz et al. 2010]
- Reproducibility [Yu 2013; Murdoch et al. 2019]
- Predictive inference with jackknife/cross-validation
  [Steinberger & Leeb 2018; B., Candès, Ramdas, Tibshirani 2021]
- Asymptotics of cross-validation
  [Kumar et al. 2013; Austern & Zhou 2020; Bayle et al. 2020]

Can algorithmic stability be assumed?

Some algorithms are stable by construction, e.g., k-NN, ridge regression

[Rogers & Wagner 1978; Bousquet & Elisseeff 2002]

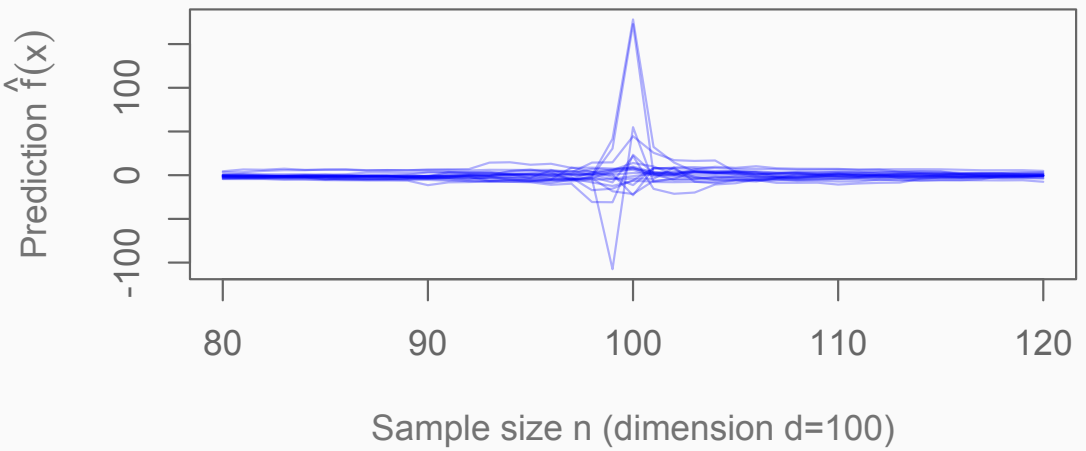
Can algorithmic stability be assumed?

Some algorithms are stable by construction, e.g., k-NN, ridge regression

[Rogers & Wagner 1978; Bousquet & Elisseeff 2002]

But, even simple algorithms can be unstable

- Least squares: $\widehat{f}(x) = x^\top \widehat{\theta}$ where $\widehat{\theta} = (\mathbf{X}^\top \mathbf{X})^\dagger \mathbf{X}^\top \mathbf{Y}$



$\leftarrow$ unstable when $n \approx d$

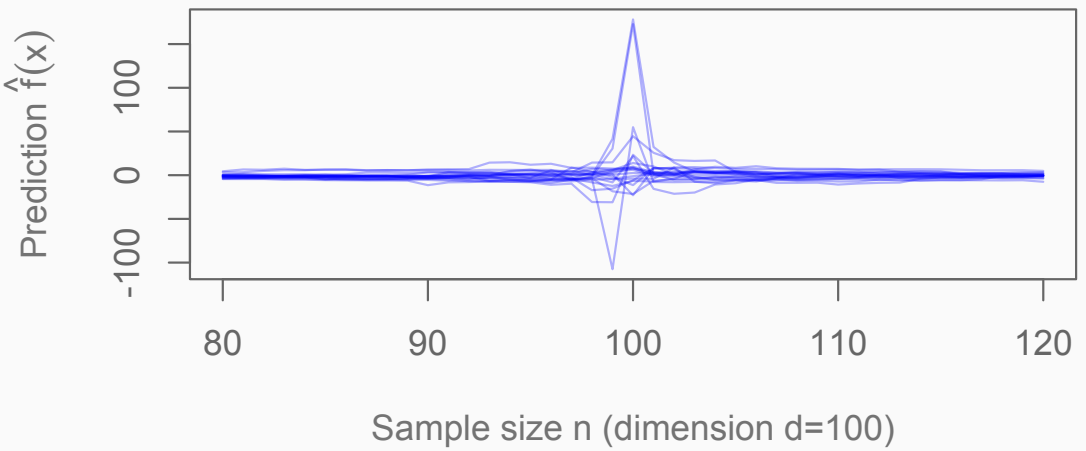
Can algorithmic stability be assumed?

Some algorithms are stable by construction, e.g., k-NN, ridge regression

[Rogers & Wagner 1978; Bousquet & Elisseeff 2002]

But, even simple algorithms can be unstable

- Least squares: $\widehat{f}(x) = x^\top \widehat{\theta}$ where $\widehat{\theta} = (\mathbf{X}^\top \mathbf{X})^\dagger \mathbf{X}^\top \mathbf{Y}$



← unstable when $n \approx d$

And, many modern algorithms are too complex for theoretical analysis

Can algorithmic stability be tested?

If stability is not known theoretically — can we certify it empirically?

The "black-box" setting:
We can only learn how $\mathcal{A}$ works by running it on data, e.g.,

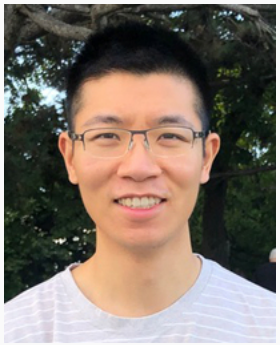
- subsamples / bootstrapped samples of the real data
- perturbations of the real data
- simulate data from a model fitted to real data

Can algorithmic stability be tested?

Hardness result for black-box tests:
To certify that $\mathcal{A}$ is stable with respect to P at sample size n,

- Need available sample size $\gg n$ drawn from P
- Or, need to perform exhaustive search over the data space

Byol Kim



Yuetian Luo

Kim & B. 2021, *Black-box tests for algorithmic stability*, arXiv:2111.15546
Luo & B. 2024, *Is algorithmic stability testable? A unified framework under computational constraints*, arXiv:2405.15107

Key question

Since algorithmic stability can't be certified empirically
(and we don't want to always use simple algorithms like k-NN) .....

Key question

Is there a way to convert any algorithm into a stable algorithm, while still retaining its good performance?

Key question

Our work uses *bagging* to address this question in a range of settings:

- Part 1: stability for regression (real-valued response)
- Part 2: stability in Hilbert & Banach spaces (e.g., functions)
- Part 3: stability for classification

Jake Soloff

Rebecca Willett

Bagging provides assumption-free stability, `arxiv:2301.12600`

Stability via resampling: statistical problems beyond the real line, `arxiv:2405.09511`

Building a stable classifier with the inflated argmax, `arxiv:2405.14064`

Part 1: stability for bagged algorithms in regression

Background on bagging

Empirically, bagging (& other ensembling procedures) have been observed to improve stability dramatically.

Bagging

- Sample subsets of data ("bags") r_b for $b = 1, \ldots, B$
- Fit models $\widehat{f}_b = \mathcal{A}(\mathcal{D}_{r_b}; \xi_b)$ where $\mathcal{D}_{r_b} = \big((X_i, Y_i) : i \in r_b\big)$

  ξ_b: random seed
- $\widetilde{\mathcal{A}}_B$ returns aggregated model $\widehat{f}$:

$$\widehat{f}(x) := \frac{1}{B} \sum_b \widehat{f}_b(x)$$

Background on bagging

The most common options:

- Bootstrapping ("bagging") = subsets r_b of size m, sampled uniformly with replacement

$$p = \mathbb{P}\{i \in r_b\} = 1 - (1 - 1/n)^m$$

- Subbagging = subsets r_b of size $m < n$, sampled uniformly without replacement

$$p = \mathbb{P}\{i \in r_b\} = m/n$$

[Breiman 1996; Andonova et al. 2002]

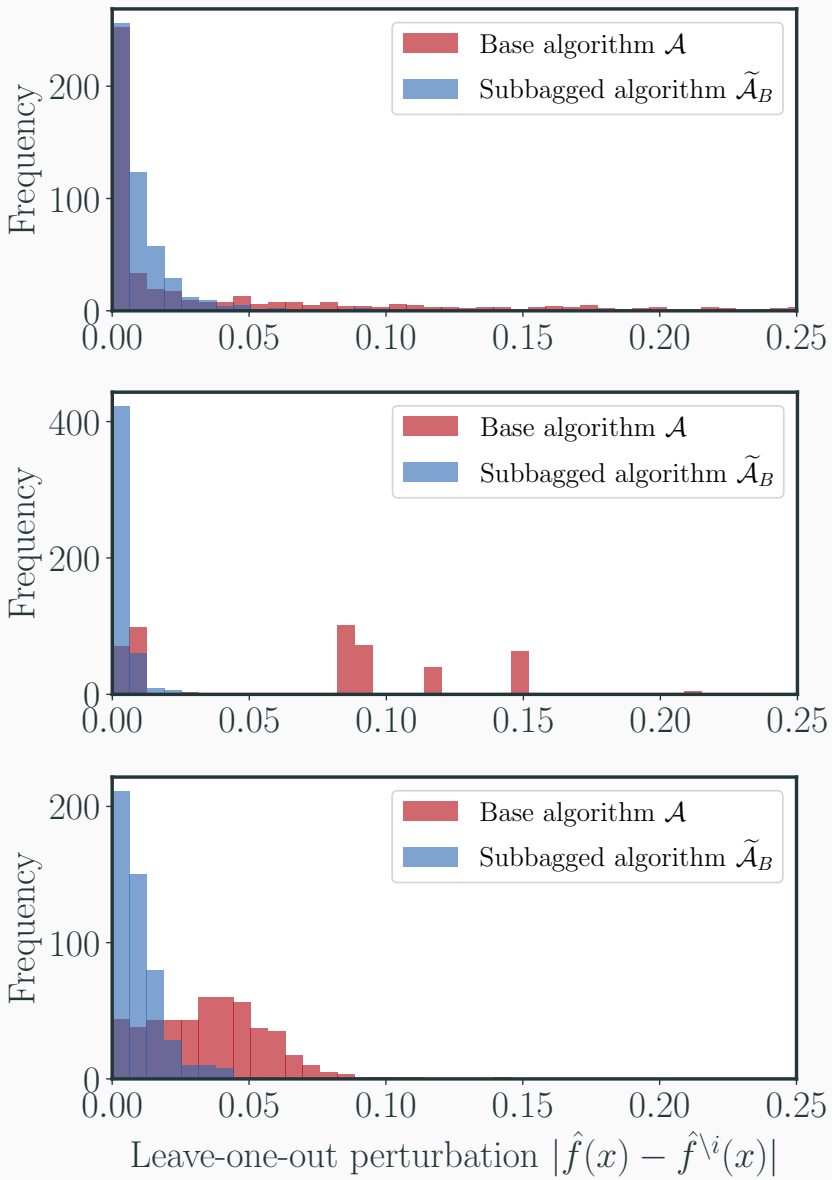
Background on bagging

Bagging appears in:

- Random forests [Breiman 2001]
- Variable selection in regression [Meinshausen & Bühlmann 2010]
- Classification in the presence of class imbalance [Li 2007]
- Robust Bayesian inference (BayesBag) [Huggins & Miller 2023]
- & many more

Many results on theoretical properties—
Bagging induces smoothness, reduces variance, creates robustness
[Bühlmann & Yu 2002; Grandvalet 2004; Friedman & Hall 2000; Elisseeff et al. 2005]

Bagging helps stability



← logistic regression

← regression trees

← neural networks

Main result: stability guarantee

Recall definition of (ϵ, δ)-stability:

$$\frac{1}{n}\sum_{i=1}^{n} \mathbb{P}_{\xi}\left\{|\widehat{f}(x) - \widehat{f}_{-i}(x)| \geq \epsilon\right\} \leq \delta$$

Main result: stability guarantee

Recall definition of (ϵ, δ)-stability:

$$\frac{1}{n}\sum_{i=1}^{n} \mathbb{P}_{\xi}\left\{|\widehat{f}(x) - \widehat{f}_{-i}(x)| \geq \epsilon\right\} \leq \delta$$

Theorem

Let $\mathcal{A}$ be any base algorithm that returns predictions in $[0, 1]$.
Then $\widetilde{\mathcal{A}}_B$ satisfies (ϵ, δ)-stability as long as

$$\epsilon^2 \delta \geq \frac{1}{4(n-1)} \cdot \frac{p}{1-p} + \mathcal{O}(B^{-1})$$

Main result: stability guarantee

- Interpretation: the bagged version of *any* base alg. is always stable:
  - No assumptions on data—holds for all $\mathcal{D}$ and all x
  - No assumptions on $\mathcal{A}$ aside from boundedness
- Can also extend to algorithms with unbounded output, via either clipping predictions, or adapting to the range of $\hat{f}$'s

Defining stability: the mean or the tail?

Tail stability

$\mathcal{A}$ is (ϵ, δ)-stable if, for any dataset $\mathcal{D}$ of size n and any x,

$$\frac{1}{n}\sum_{i=1}^{n}\mathbb{P}_{\xi}\left\{\left|\widehat{f}(x)-\widehat{f}_{-i}(x)\right|\geq\epsilon\right\}\leq\delta$$

Mean-square stability

$\mathcal{A}$ is β^2-stable if, for any dataset $\mathcal{D}$ of size n and any x,

$$\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_{\xi}\left[\left(\widehat{f}(x)-\widehat{f}_{-i}(x)\right)^{2}\right]\leq\beta^{2}$$

Defining stability: the mean or the tail?

$$\underbrace{\beta^2\text{-stability}}_{\text{mean-square stability}} \quad \text{vs} \quad \underbrace{(\epsilon, \delta)\text{-stability}}_{\text{tail stability}}$$

- β^2-stability $\implies$ (ϵ, δ)-stability for any $\epsilon^2\delta \geq \beta^2$
- (ϵ, δ)-stability, & outputs $\in [0, 1] \implies$ β^2-stab. for $\beta^2 = \epsilon^2 + \delta$

Main result: stability guarantee

Theorem

Let $\mathcal{A}$ be any base algorithm that returns predictions in $[0, 1]$.
Then $\widetilde{\mathcal{A}}_B$ satisfies β^2-stability for

$$\beta^2 = \frac{1}{4(n-1)} \cdot \frac{p}{1-p} + \mathcal{O}(B^{-1})$$

- Directly implies the (ϵ, δ)-stability result

Main result: stability guarantee

Regimes for subbagging (with $B \to \infty$):

- **Proportional sampling:**

$$m = \mathcal{O}(n) \implies \beta^2 \propto n^{-1}$$

Main result: stability guarantee

Regimes for subbagging (with $B \to \infty$):

- **Proportional sampling:**

$$m = \mathcal{O}(n) \implies \beta^2 \propto n^{-1}$$

- **Massive subsampling:** $0 < a < 1$

$$m = \mathcal{O}(n^a) \implies \beta^2 \propto n^{-(2-a)}$$

[See also Poggio et al. 2002; Elisseeff et al. 2005; Chen et al. 2022]

Main result: stability guarantee

Regimes for subbagging (with $B \to \infty$):

- **Proportional sampling:**

$$m = \mathcal{O}(n) \implies \beta^2 \propto n^{-1}$$

- **Massive subsampling:** $0 < a < 1$

$$m = \mathcal{O}(n^a) \implies \beta^2 \propto n^{-(2-a)}$$

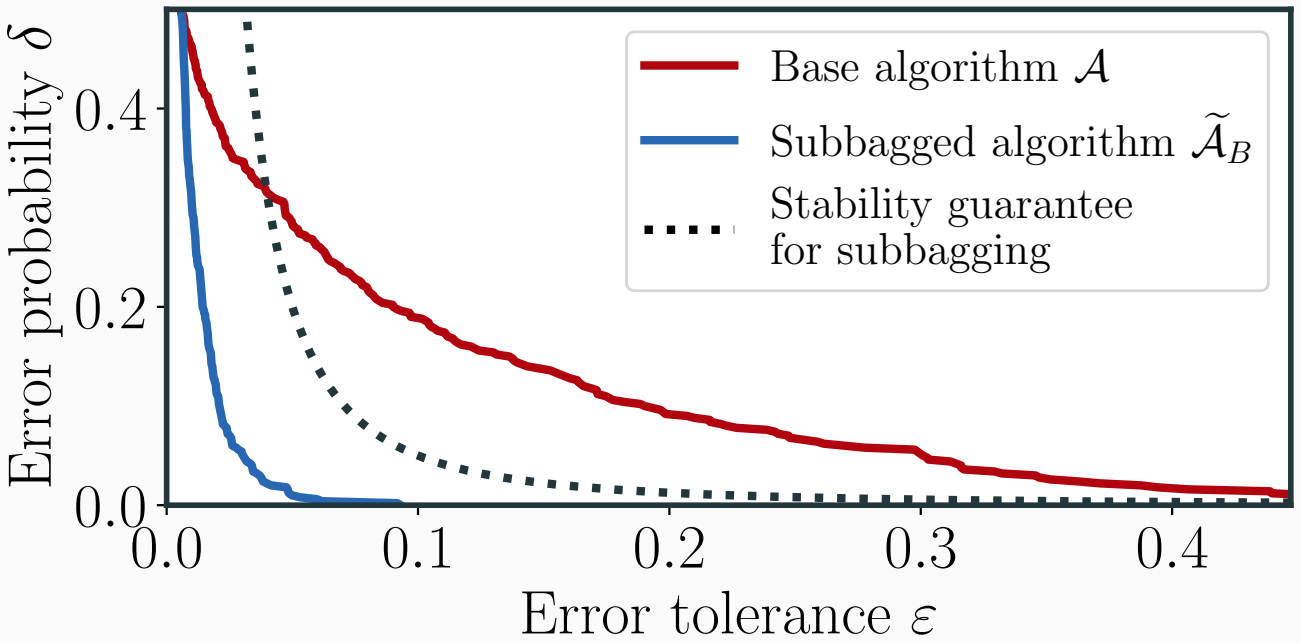
[See also Poggio et al. 2002; Elisseeff et al. 2005; Chen et al. 2022]

- **Minimal subsampling:**

$$m = n - \mathcal{O}(n^a) \implies \beta^2 \propto n^{-a}$$

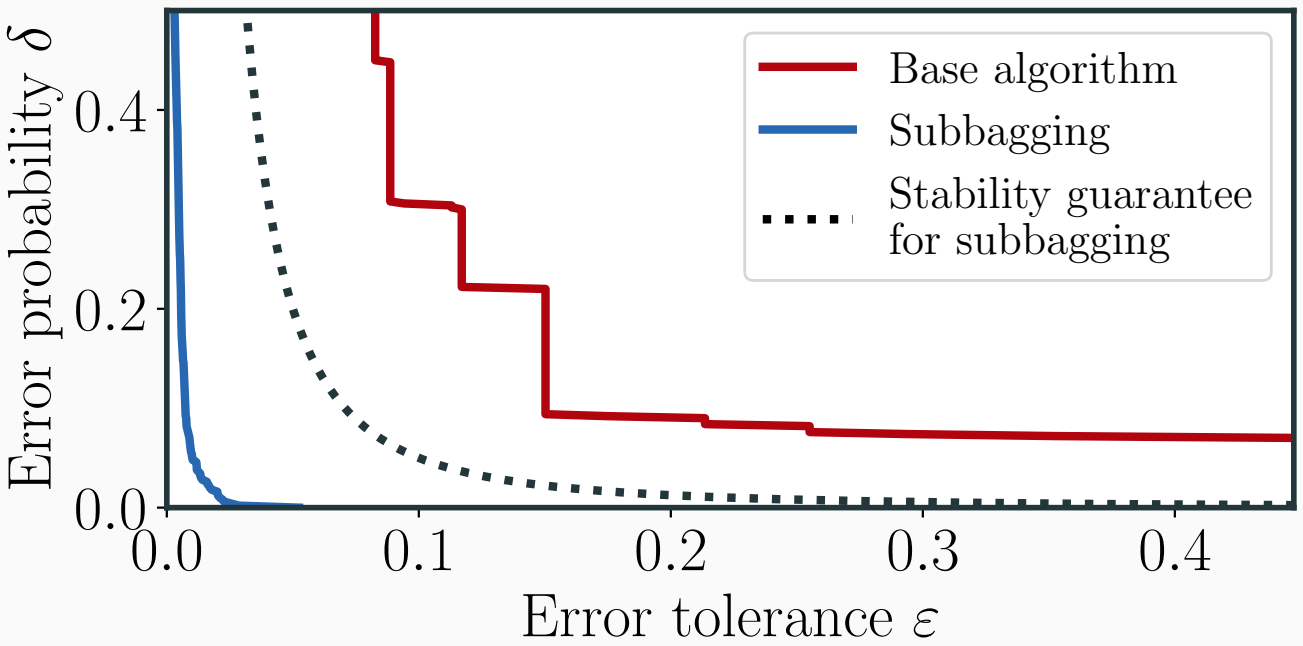
Experiments

Logistic regression ($d = 200$, $n = 500$, $m = n/2$, $B = 10000$)



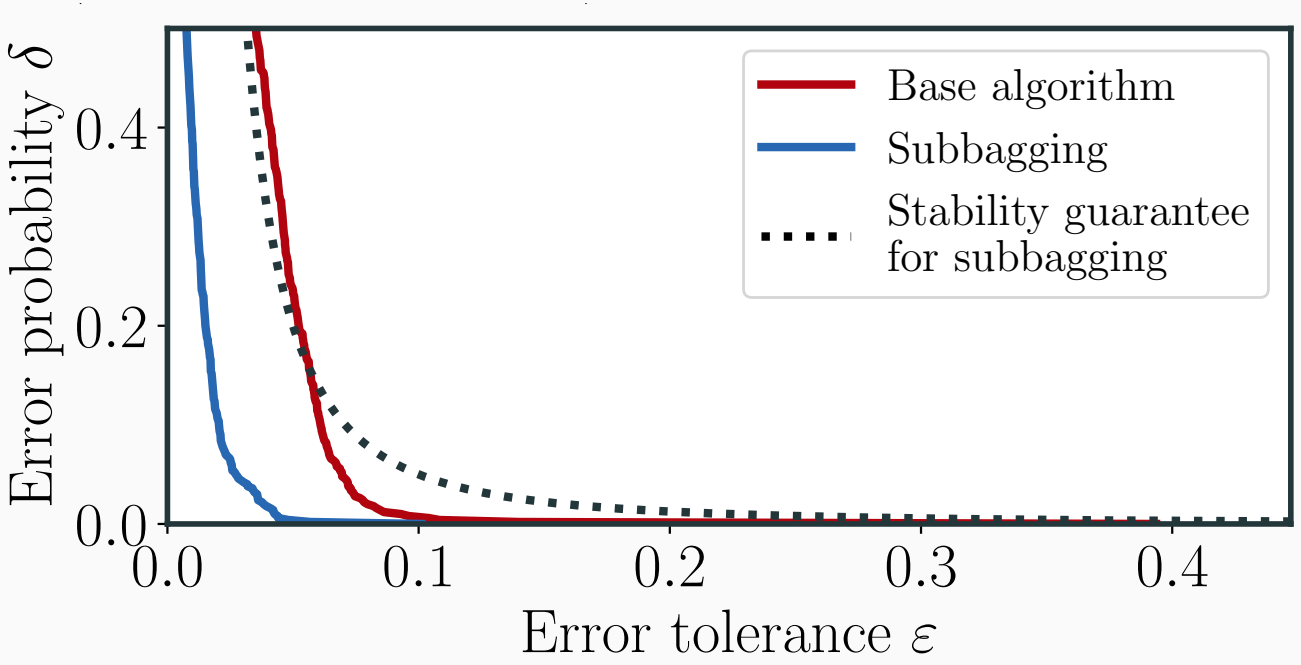
Experiments

Regression trees ($d = 40$, $n = 500$, $m = n/2$, $B = 10000$)



Experiments

Neural networks ($d = 200$, $n = 500$, $m = n/2$, $B = 10000$)



Is the guarantee tight?

Could the bound be improved?

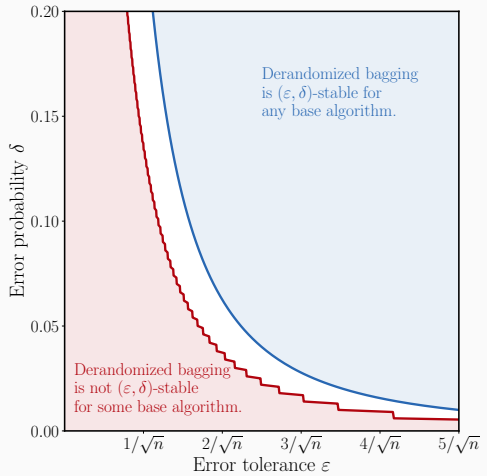
Theorem: a matching bound for subbagging

Fix n, p with $\min\{p, 1-p\} \gg \frac{1}{n}$.
There exists an $\mathcal{A}$ with output in $[0,1]$, such that if

$$\beta^2 < \frac{1}{c(n-1)} \cdot \frac{p}{1-p}$$

then $\widetilde{\mathcal{A}}_\infty$ is *not* β^2-stable.

Illustration for $n = 500$, $m = 250$:



Is the guarantee tight?

Could we use a stricter definition of stability?

Worst-case stability

$\mathcal{A}$ is β-worst-case-stable if, for any dataset $\mathcal{D}$ of size n and any x,

$$\max_i \left| \widehat{f}(x) - \widehat{f}_{-i}(x) \right| \leq \beta$$

max instead of average

Is the guarantee tight?

Could we use a stricter definition of stability?

Worst-case stability

$\mathcal{A}$ is β-worst-case-stable if, for any dataset $\mathcal{D}$ of size n and any x,

$$\max_i \left| \widehat{f}(x) - \widehat{f}_{-i}(x) \right| \leq \beta$$

max instead of average

- For any $\mathcal{A}$ with output in $[0, 1]$, $\widetilde{\mathcal{A}}_\infty$ is β-worst-case-stable for $\beta = p$
  [Poggio et al. 2002; Elisseeff et al. 2005; Chen et al. 2022]
- For any $\beta < p$, can construct $\mathcal{A}$ s.t. $\widetilde{\mathcal{A}}_\infty$ is not β-worst-case-stable

Part 2: bagged algorithms beyond the real line

Settings & applications

The stability result proved so far....

- Stated for prediction: stability of $\mathcal{D} \mapsto [\mathcal{A}(\mathcal{D})](x) \in \mathbb{R}$
- Result holds for *any* real-valued output, i.e., $\mathcal{D} \mapsto \mathcal{A}(\mathcal{D}) \in \mathbb{R}$
  (doesn't have to be "a prediction")

Settings & applications

The stability result proved so far....

- Stated for prediction: stability of $\mathcal{D} \mapsto [\mathcal{A}(\mathcal{D})](x) \in \mathbb{R}$
- Result holds for *any* real-valued output, i.e., $\mathcal{D} \mapsto \mathcal{A}(\mathcal{D}) \in \mathbb{R}$
  (doesn't have to be "a prediction")

Many data analysis algorithms return output that is more complex:

- Estimate a discrete distribution $\rightsquigarrow \mathcal{A}(\mathcal{D}) \in \Delta_{L-1} \subseteq \mathbb{R}^L$ (simplex)
- Estimate a density on $\mathbb{R} \rightsquigarrow \mathcal{A}(\mathcal{D}) \in L_1(\mathbb{R})$
- Fit a parametric model $\rightsquigarrow \mathcal{A}(\mathcal{D}) \in \Theta \subseteq \mathbb{R}^d$

Stability guarantee for a Hilbert space

Suppose $\mathcal{A}$ returns outputs lying in a Hilbert space:

$$\mathcal{A} : \bigcup_{n \geq 0} \mathcal{Z}^n \longrightarrow \mathcal{W} \subseteq \mathcal{H}$$

Hilbert space with norm $\|\cdot\|_{\mathcal{H}}$

Stability guarantee for a Hilbert space

Suppose $\mathcal{A}$ returns outputs lying in a Hilbert space:

$$\mathcal{A} : \bigcup_{n \geq 0} \mathcal{Z}^n \longrightarrow \mathcal{W} \subseteq \mathcal{H}$$

Hilbert space with norm $\|\cdot\|_{\mathcal{H}}$

Assume $\mathcal{W}$ is a bounded subset:

$$\mathrm{rad}_{\mathcal{H}}(\mathcal{W}) = \inf_{w \in \mathcal{W}} \sup_{w' \in \mathcal{W}} \|w - w'\|_{\mathcal{H}}$$

- Example: $\mathcal{W} = \Delta_{L-1} \subseteq \mathbb{R}^L \rightsquigarrow \mathrm{rad}_{\mathcal{H}}(\mathcal{W})^2 = 1 - \frac{1}{L}$

Stability guarantee for a Hilbert space

Definition

$\mathcal{A}$ is β^2-stable with respect to $\|\cdot\|_{\mathcal{H}}$ if, for any dataset $\mathcal{D}$ of size n,

$$\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_{\xi}\left[\ \|\mathcal{A}(\mathcal{D};\xi)-\mathcal{A}(\mathcal{D}_{-i};\xi)\|_{\mathcal{H}}^{2}\ \right]\leq\beta^{2}$$

Theorem

Let $\mathcal{A}$ be any base algorithm that returns predictions in $\mathcal{W}\subseteq\mathcal{H}$.
Then $\widetilde{\mathcal{A}}_B$ satisfies β^2-stability with respect to $\|\cdot\|_{\mathcal{H}}$ for

$$\beta^{2}=\frac{\mathrm{rad}_{\mathcal{H}}(\mathcal{W})^{2}}{n-1}\cdot\frac{p}{1-p}+\mathcal{O}(B^{-1}).$$

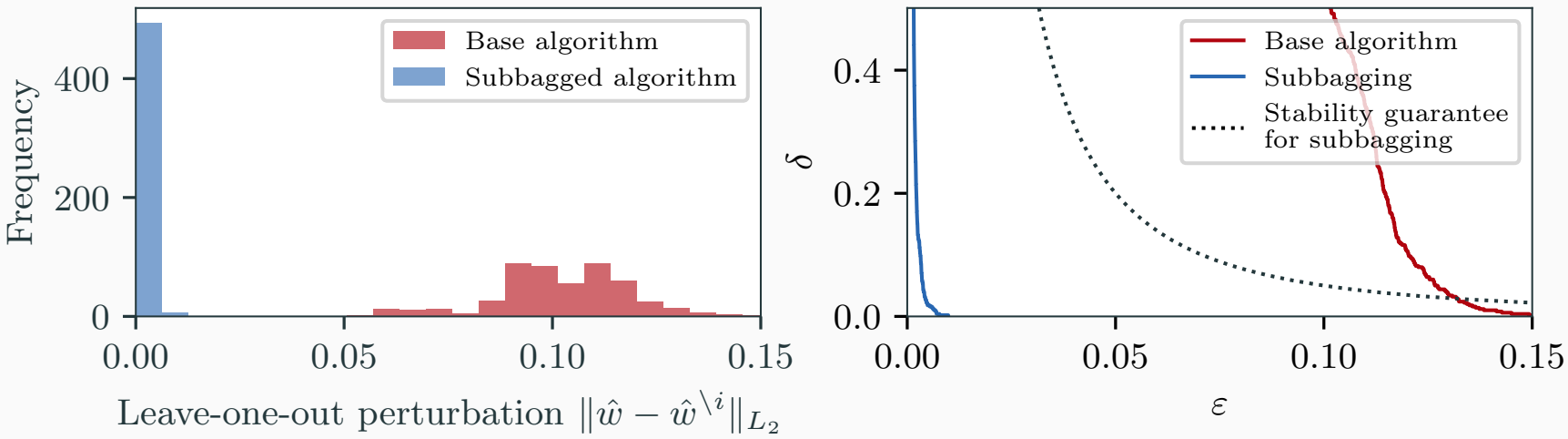
Stability guarantee for a Hilbert space

- No price for high dimensionality
- Earlier result for $[0,1] \subseteq \mathbb{R}$ is a special case, with $\mathrm{rad}_{\mathcal{H}}(\mathcal{W}) = 1/2$
- Can extend to Banach space (e.g., d_{TV} distance) $\rightsquigarrow$ log(dim) terms

Experiments

Estimating a regression function in L_2 (base algorithm = regression tree)

Setting: $d = 40$, $n = 500$, $m = n/2$, $B = 10000$



Part 3: a stability framework for classification

Can classification be stable?

Returning to the supervised learning setting...

- Training data $\mathcal{D} = \big((X_1, Y_1), \ldots, (X_n, Y_n)\big)$
- Test point x
- $\mathcal{Y} = [L]$ = finite set of labels

If we use $\mathcal{D}$ to train a classifier (a map $x \mapsto \widehat{y} \in [L]$)...

- Instability must occur b/c $\widehat{y} \neq \widehat{y}_{-i}$ is likely for ambiguous examples

Can classification be stable?

Returning to the supervised learning setting...

- Training data $\mathcal{D} = \big((X_1, Y_1), \ldots, (X_n, Y_n)\big)$
- Test point x
- $\mathcal{Y} = [L]$ = finite set of labels

If we use $\mathcal{D}$ to train a classifier (a map $x \mapsto \widehat{y} \in [L]$)...

- Instability must occur b/c $\widehat{y} \neq \widehat{y}_{-i}$ is likely for ambiguous examples
- A common approach: return a set of candidate labels, $x \mapsto \widehat{S} \subseteq [L]$

  [Grycko 1993; del Coz et al. 2009; Sadinle et al. 2019; Chzhen et al. 2021]

Can classification be stable?

Selection stability

A classification algorithm satisfies selection stability at level δ if

$$\frac{1}{n}\sum_{i=1}^{n}\mathbb{1}\left\{\widehat{S}\cap\widehat{S}_{-i}=\varnothing\right\}\leq\delta$$

where $\widehat{S}=[\mathcal{A}(\mathcal{D})](x)$ and $\widehat{S}_{-i}=[\mathcal{A}(\mathcal{D}_{-i})](x)$

Interpretation: if $\widehat{S}\cap\widehat{S}_{-i}\neq\varnothing$, then it's possible to have

$$y_*\in\widehat{S}\text{ and }y_*\in\widehat{S}_{-i}$$

where y_* is the true label

Can classification be stable?

Selection stability

A classification algorithm satisfies selection stability at level δ if

$$\frac{1}{n}\sum_{i=1}^{n}\mathbb{P}_{\xi}\left\{\widehat{S}\cap\widehat{S}_{-i}=\varnothing\right\}\leq\delta$$

where $\widehat{S}=[\mathcal{A}(\mathcal{D};\xi)](x)$ and $\widehat{S}_{-i}=[\mathcal{A}(\mathcal{D}_{-i};\xi)](x)$

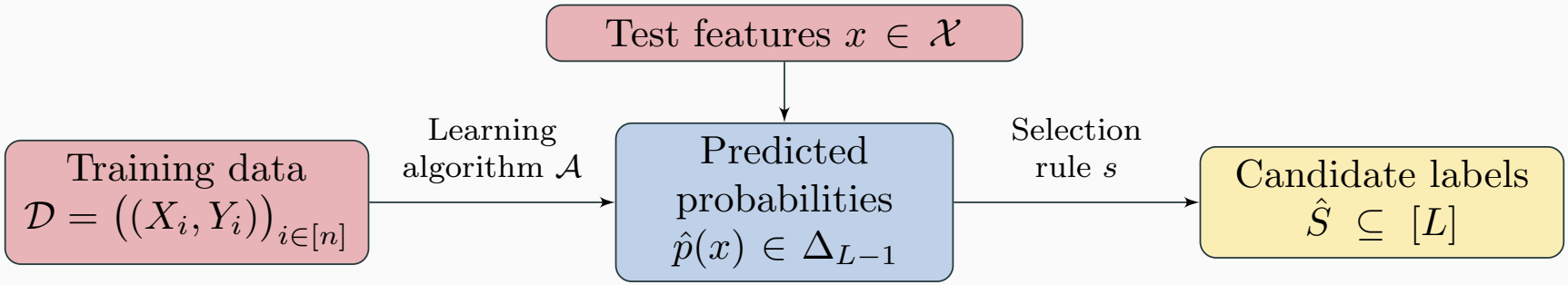
Interpretation: if $\widehat{S}\cap\widehat{S}_{-i}\neq\varnothing$, then it's possible to have

$$y_{*}\in\widehat{S}\text{ and }y_{*}\in\widehat{S}_{-i}$$

where y_{*} is the true label

A two-stage framework

Building a classification algorithm:



A two-stage framework

Definition: ϵ-compatible selection rule

If $w, w' \in \Delta_{L-1}$ with $\|w - w'\|_2 < \epsilon$, then $s(w) \cap s(w') \neq \varnothing$

Stability in the two-stage framework

Any (ϵ, δ)-stable $\mathcal{A}$ $+$ any ϵ-compatible s $\Rightarrow$ δ-selection-stable $s \circ \mathcal{A}$

A two-stage framework

Definition: ϵ-compatible selection rule

If $w, w' \in \Delta_{L-1}$ with $\|w - w'\|_2 < \epsilon$, then $s(w) \cap s(w') \neq \varnothing$

Stability in the two-stage framework

Any (ϵ, δ)-stable $\mathcal{A}$ + any ϵ-compatible s $\Rightarrow$ δ-selection-stable $s \circ \mathcal{A}$

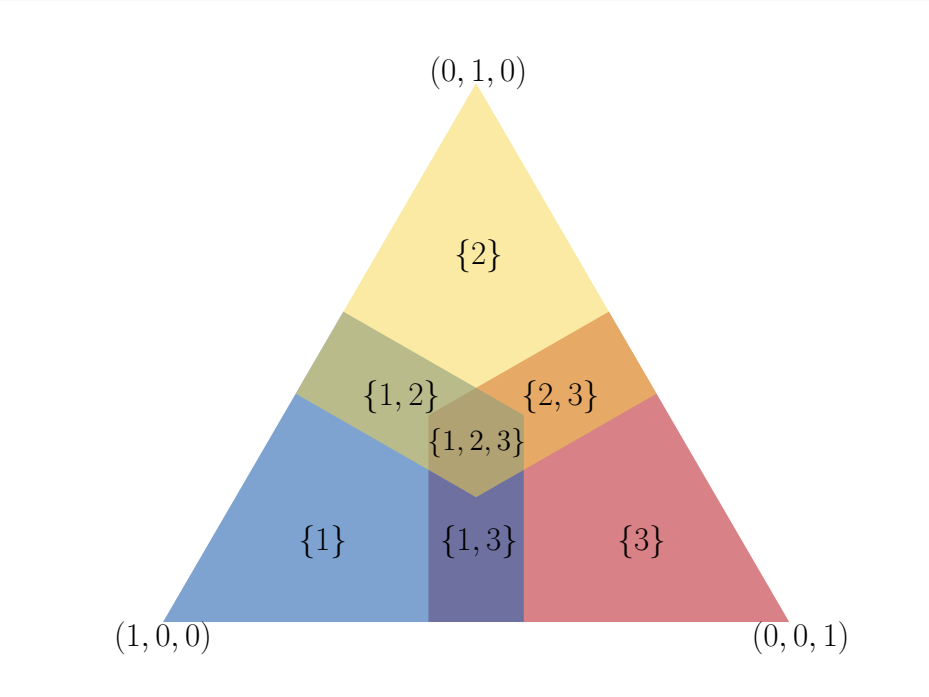
construct via bagging any base alg.

how to define s?

A baseline: the fixed-margin rule

A fixed-margin selection rule is ϵ-compatible:

$$s^{\epsilon}_{\text{margin}}(w) = \left\{ j \in [L] : w_j > \max_k w_k - \epsilon/\sqrt{2} \right\}$$



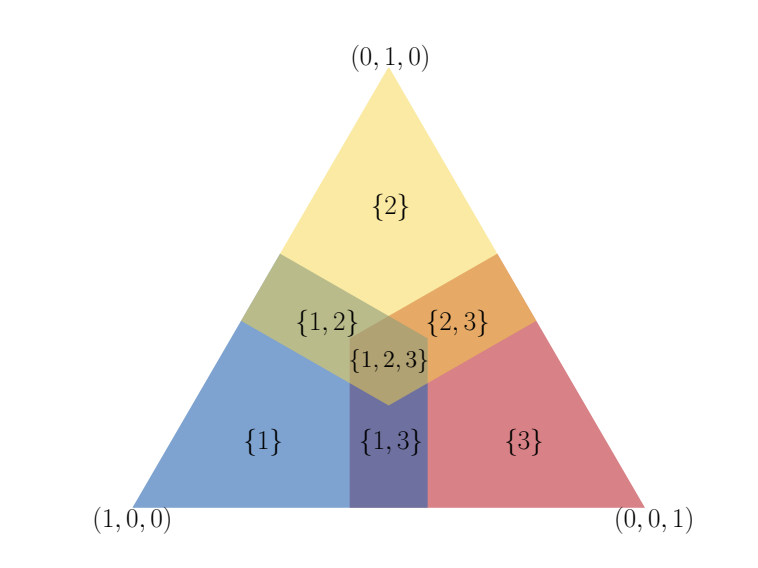
The inflated argmax

Is the fixed-margin rule optimal?
(Does it return small sets whenever possible?)

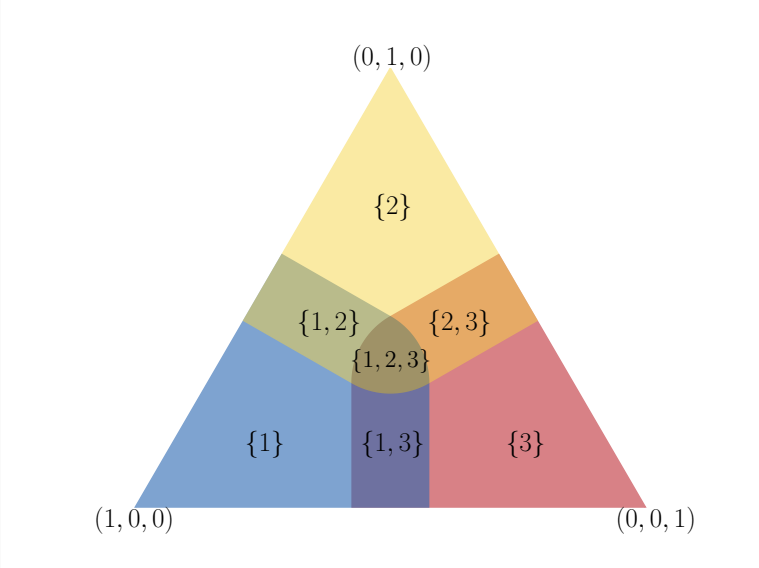
The inflated argmax

Is the fixed-margin rule optimal?
(Does it return small sets whenever possible?)

Comparison:



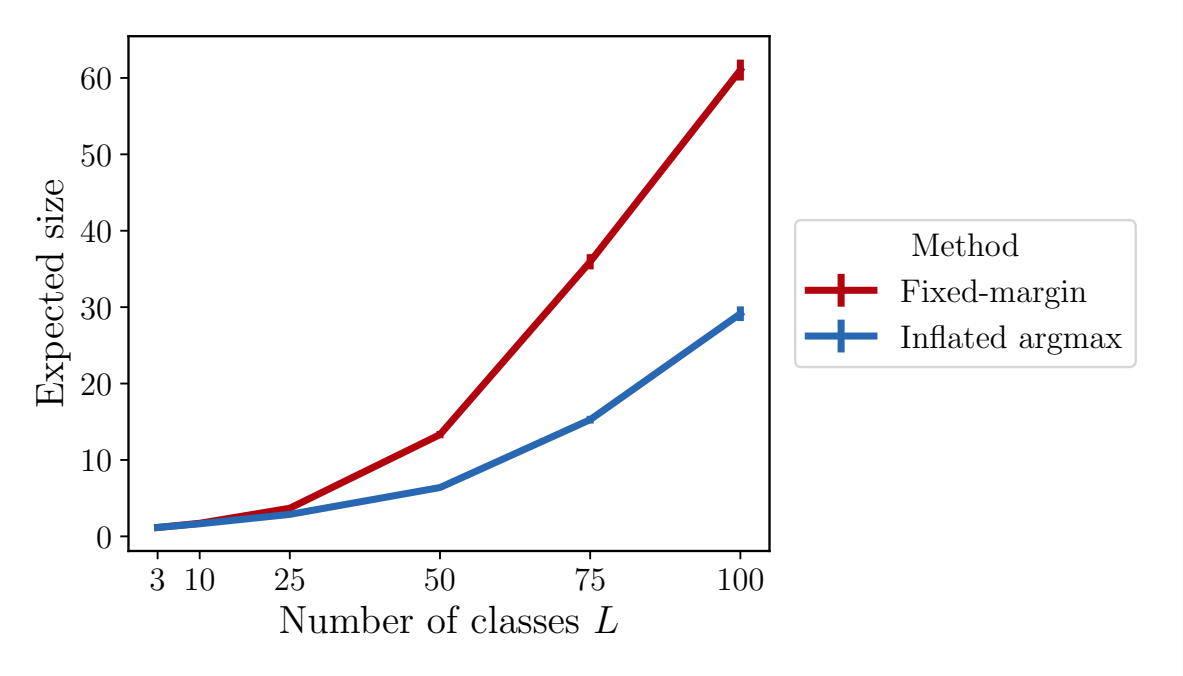
Fixed-margin rule



Inflated argmax

The inflated argmax

Another comparison:



The inflated argmax

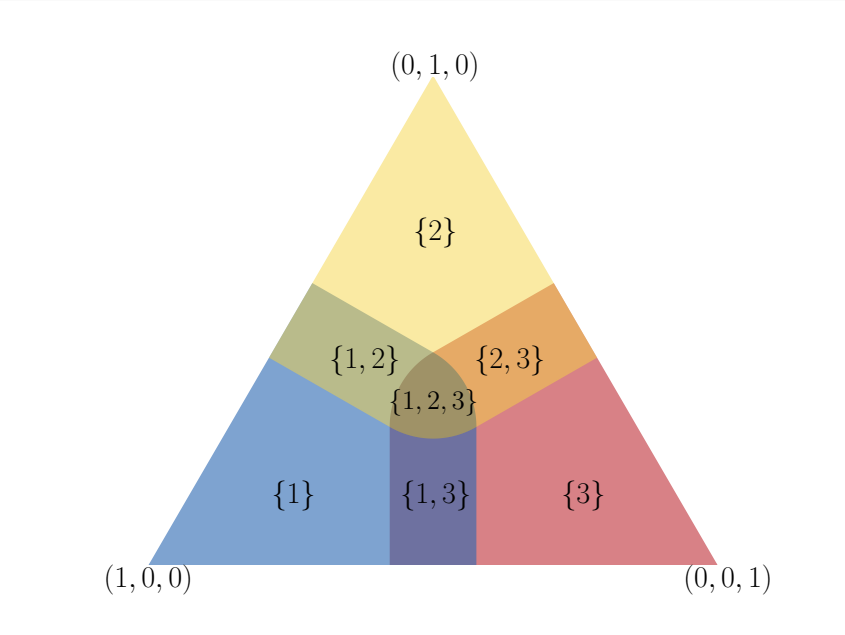
The inflated argmax selection rule

Define

$$\arg\max^{\epsilon}(w) = \left\{ j \in [L] : \mathrm{dist}_{\ell_2}(w, R_j) < \epsilon \right\},$$

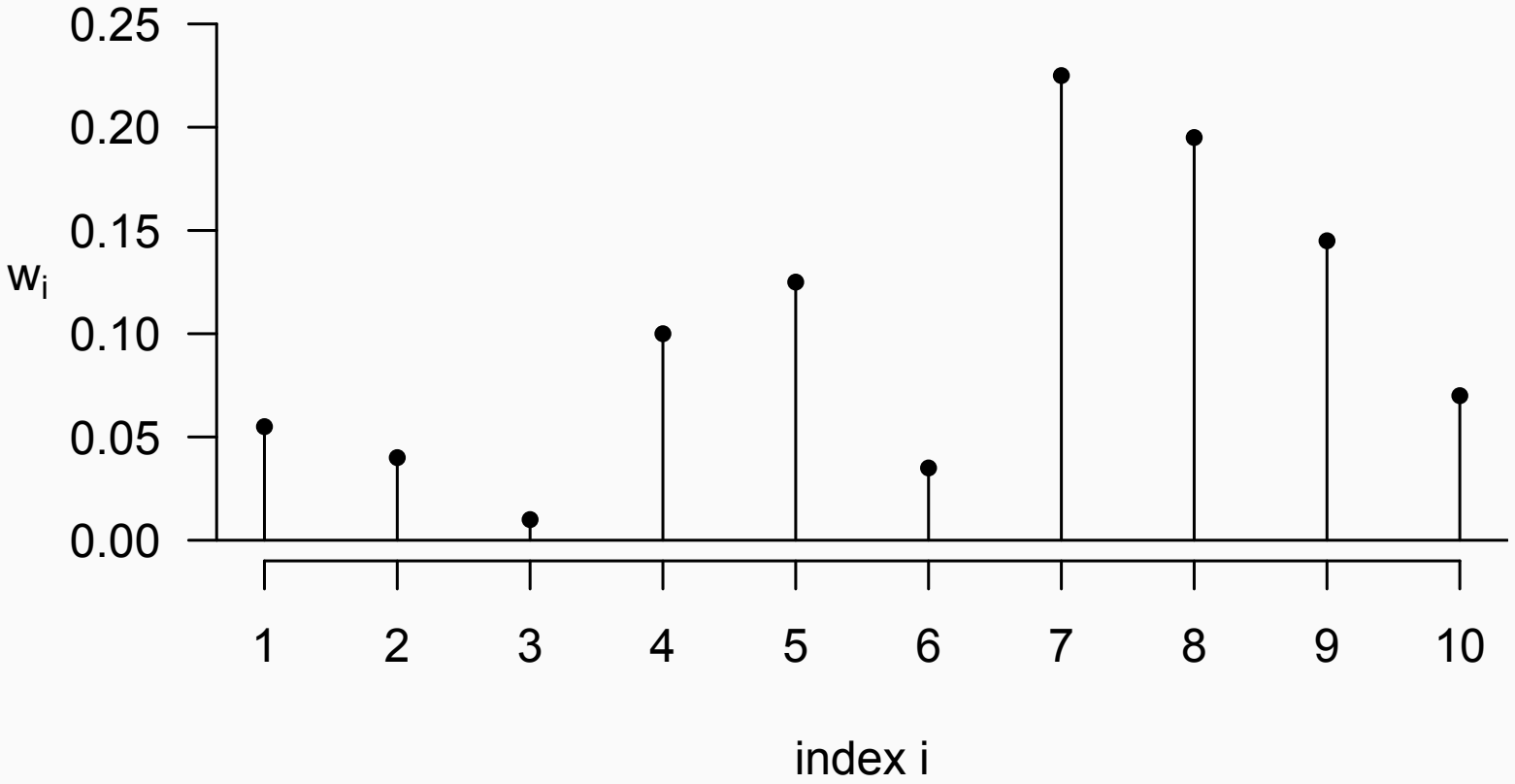
where

$$R_j = \left\{ w \in \mathbb{R}^L : w_j \geq \max_{k \neq j} w_k + \epsilon/\sqrt{2} \right\}$$



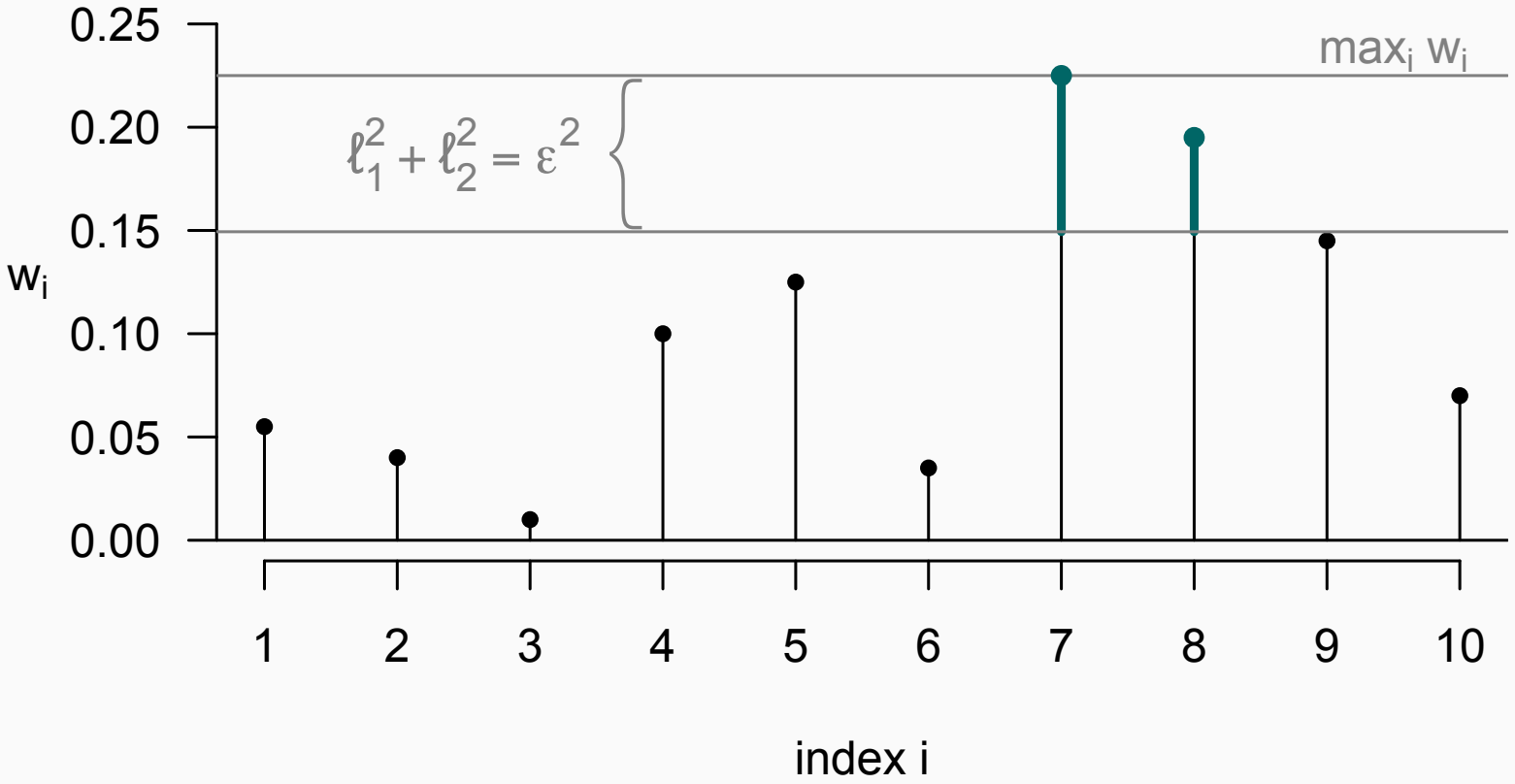
The inflated argmax

Computing the inflated argmax:



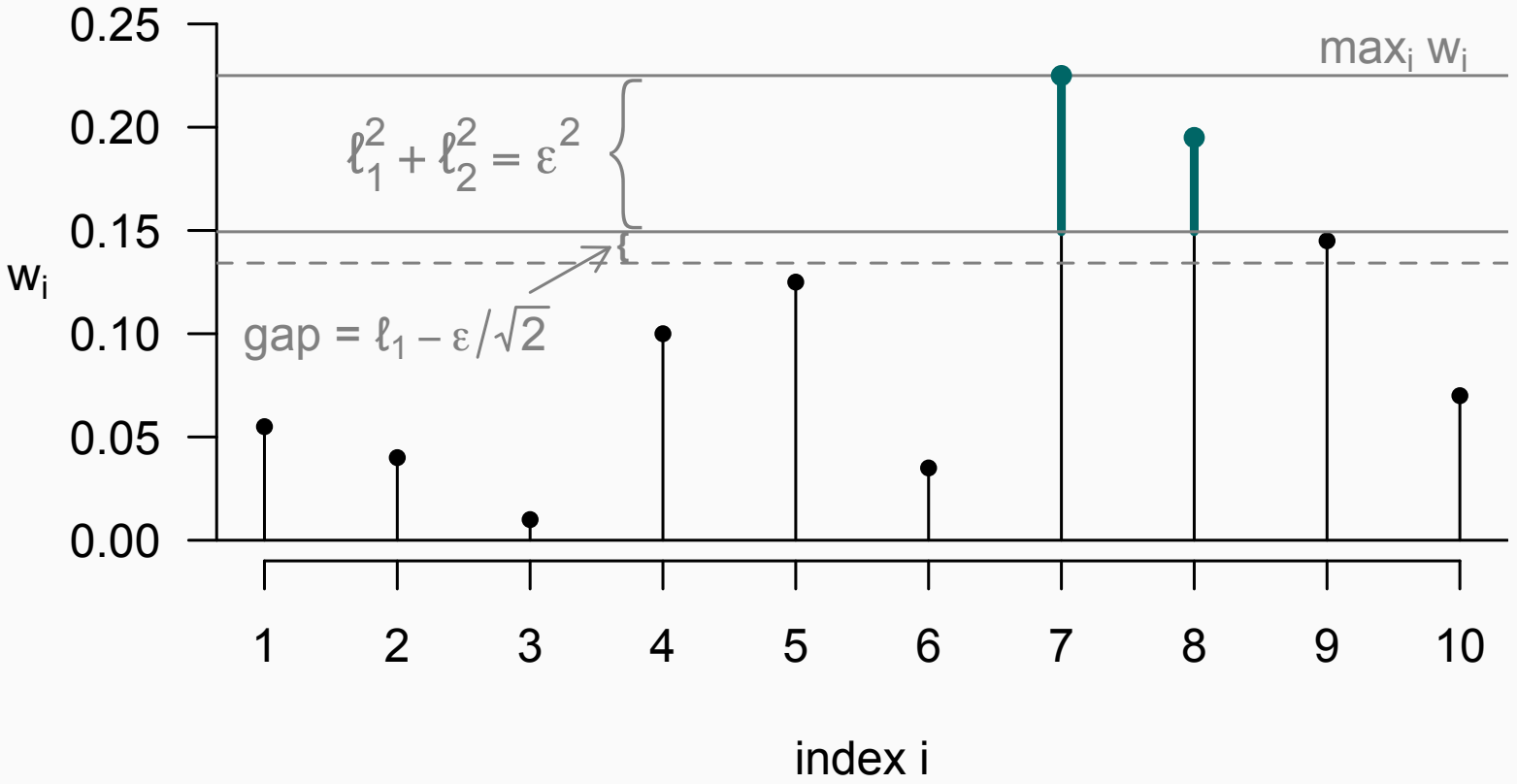
The inflated argmax

Computing the inflated argmax:



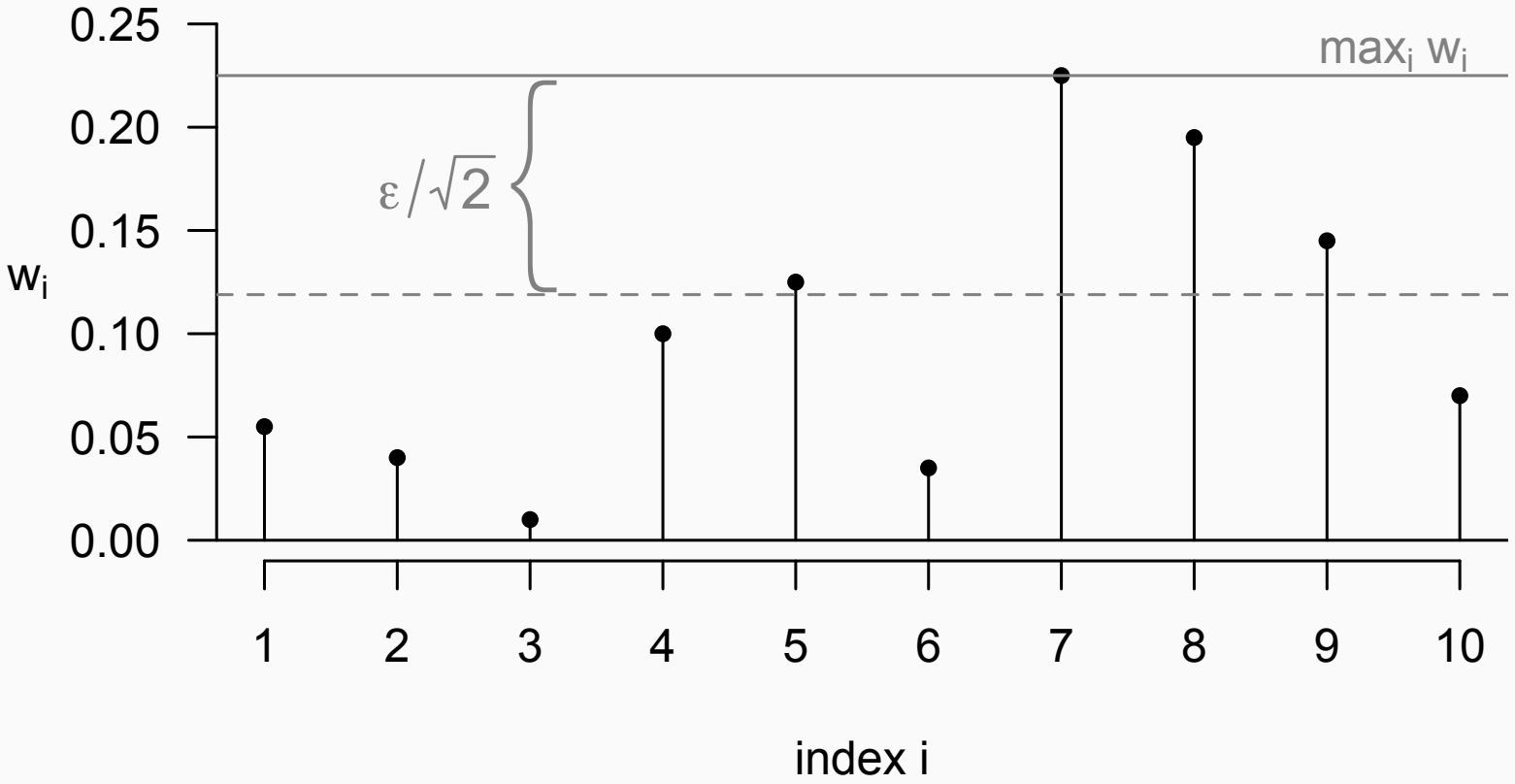
The inflated argmax

Computing the inflated argmax:



The inflated argmax

Compare to the fixed-margin selection rule:



Properties of the inflated argmax

Theorem (compatibility)

The inflated argmax is ϵ-compatible:

$$\|w - w'\|_2 < \epsilon \implies \arg\max^{\epsilon}(w) \cap \arg\max^{\epsilon}(w') \neq \varnothing$$

Interpretation: any bagged alg. $+$ inflated argmax $\rightsquigarrow$ selection stability

Properties of the inflated argmax

Theorem (compatibility)

The inflated argmax is ϵ-compatible:

$$\|w - w'\|_2 < \epsilon \implies \arg\max^{\epsilon}(w) \cap \arg\max^{\epsilon}(w') \neq \varnothing$$

Interpretation: any bagged alg. + inflated argmax $\rightsquigarrow$ selection stability

Theorem (optimality)

For any ϵ-compatible selection rule s,
if s is symmetric and $s(w) \supseteq \arg\max(w)$ for all w,

$$s(w) = \{j\} \implies \arg\max^{\epsilon}(w) = \{j\}$$

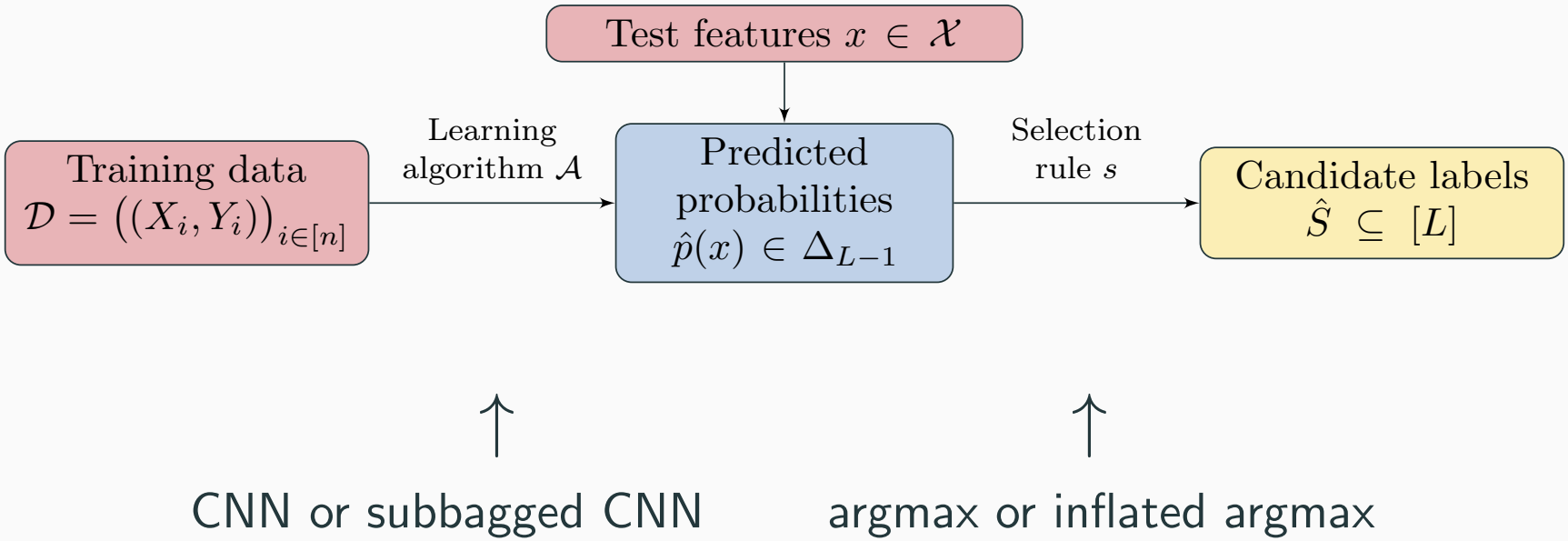
Interpretation: inflated argmax returns a singleton set as often as possible

Experiments

Dataset: fashion MNIST ($L = 10$ classes) [figure & data from Xiao et al. 2017]

Experiments

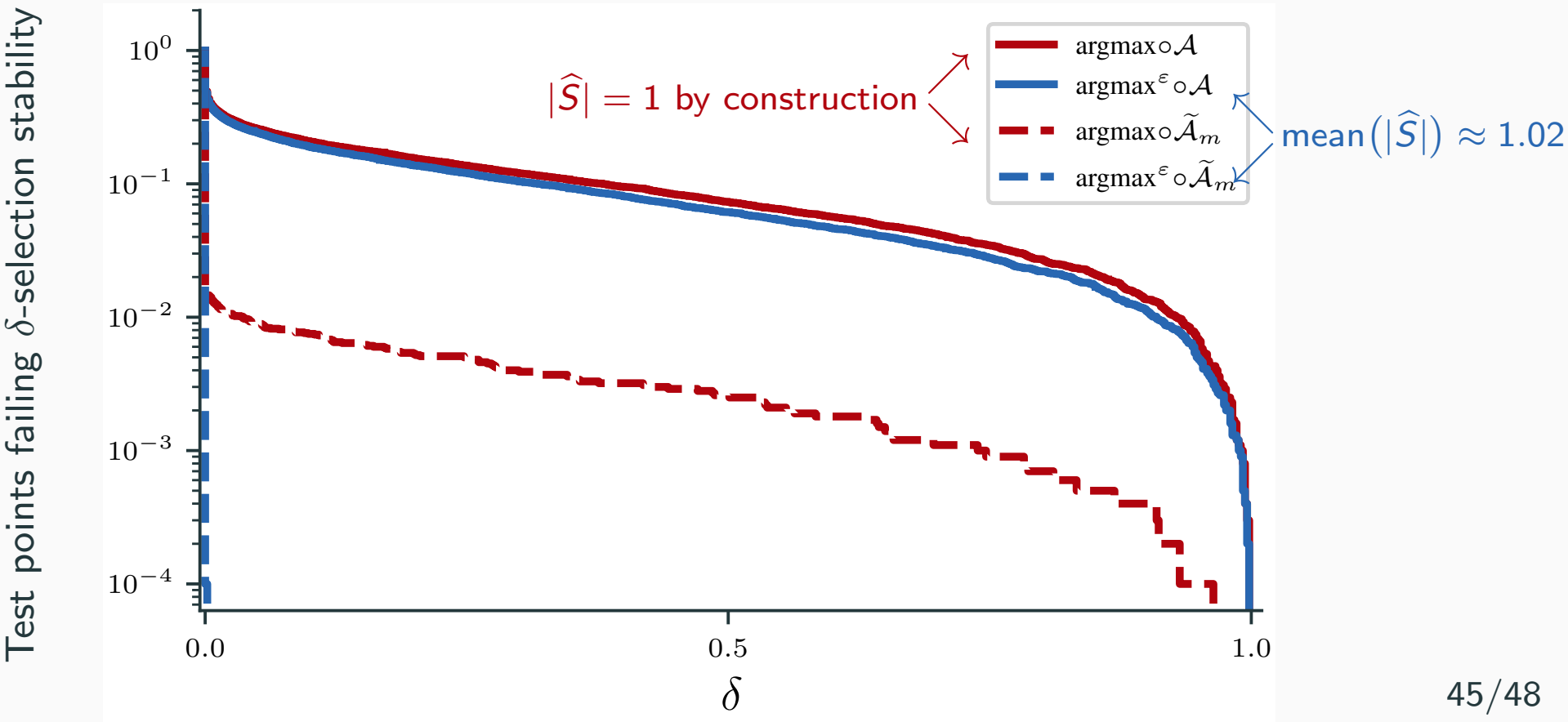
Methods tested:



Experiments

Setting: $n = 60000$, $m = n/2$, $B = 200$, inflation $\epsilon = 0.05$

Methods: {CNN or subbagged CNN} + {argmax or inflated argmax}



An extension: stability for ranking problems

Data from n users $\rightsquigarrow$ scores $\hat{w} = (\hat{w}_1, \ldots, \hat{w}_L)$ for L items.

- What are the top k items?
  (return an inflated set of size $\geq k$)
- What is the full ranked list?
  (return ≥ 1 rankings of the list)

	Godfather	The Matrix	Toy Story	...
User #1		5	4	...
User #2	3			...
User #3		1	3	...
⋮	⋮	⋮	⋮	

Ruiting Liang

Jake Soloff

Rebecca Willett

Liang, Soloff, B., & Willett 2025, *Assumption-free stability for ranking problems*, arXiv:2506.02257

Summary & open questions

Summary & open questions

Our results:

- Classical bagging & subbagging can be applied to *any* algorithm $\mathcal{A}$ to achieve an assumption-free stability guarantee
- Downstream, leads to generalization, predictive inference, etc
- Extends to outputs in a Hilbert or Banach space
  (e.g., output can be a vector, a function, a distribution, ...)
- Can combine with inflated argmax for stable classification

Summary & open questions

Open questions:

- How does bagging perform relative to other definitions of stability, & related properties — generalization, robustness, privacy, etc?
- Guarantees for aggregation procedures aside from averaging?
- Is the computational cost of bagging a necessary cost for stability?

Thank you!