Large-width generalization error bounds

Dept. of Mathematics, University of Pisa

May 22, 2025

Joel Ratsaby

Ariel University

Introduction

- Domain X
- $Y = \{-1, +1\}$
- $\mathbb{I}\{A\}$ the indicator function, for true/false statement A
- Denote $Z = X \times Y$

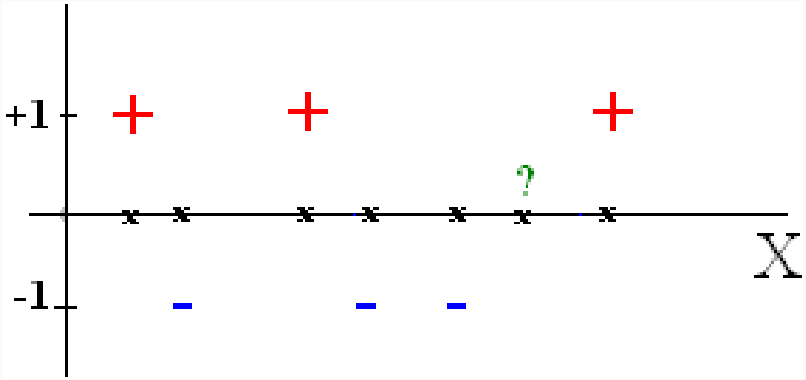
Learn to Predict/Classify

- Fix (unknown) probability distribution P on Z
- *Input*: a *random* training sample

$$\zeta = \{(x_i, y_i)\}_{i=1}^{m} \in Z^m$$

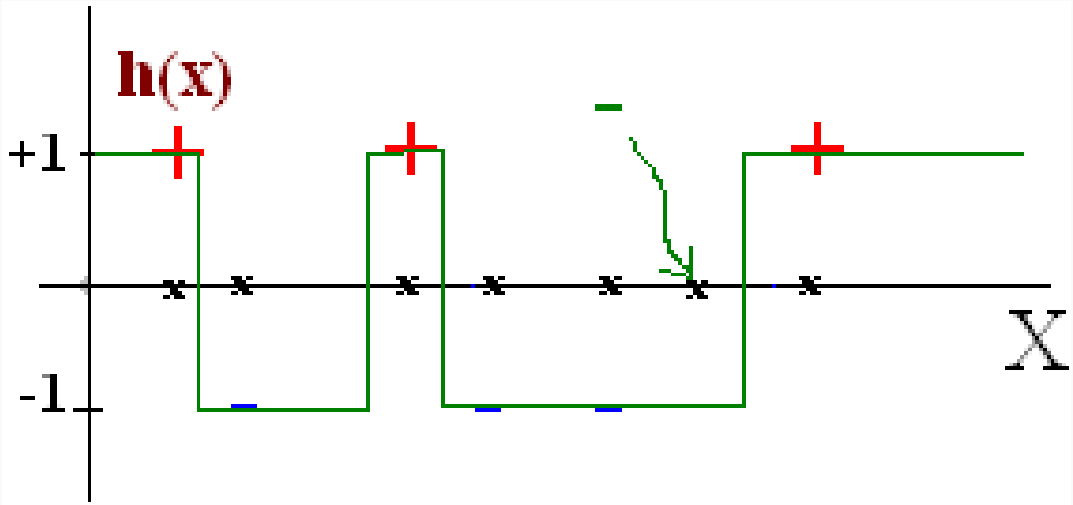
distributed according to P^m

- *Aim*: Predict the label y from x for an element (x, y) of Z which is drawn according to P



Need to generalize

- Generalize *beyond* the sample ζ
- Can use a *binary function* h on X as hypothesis
- h maps from X to $Y = \{-1, +1\}$ (discrete function)



Generalization Error

- For any binary function h, the generalization error of h is defined as

$$\begin{aligned} L(h) &= P\{h(x) \neq y\} \\ &= P\{y\, h(x) < 0\}, \end{aligned}$$

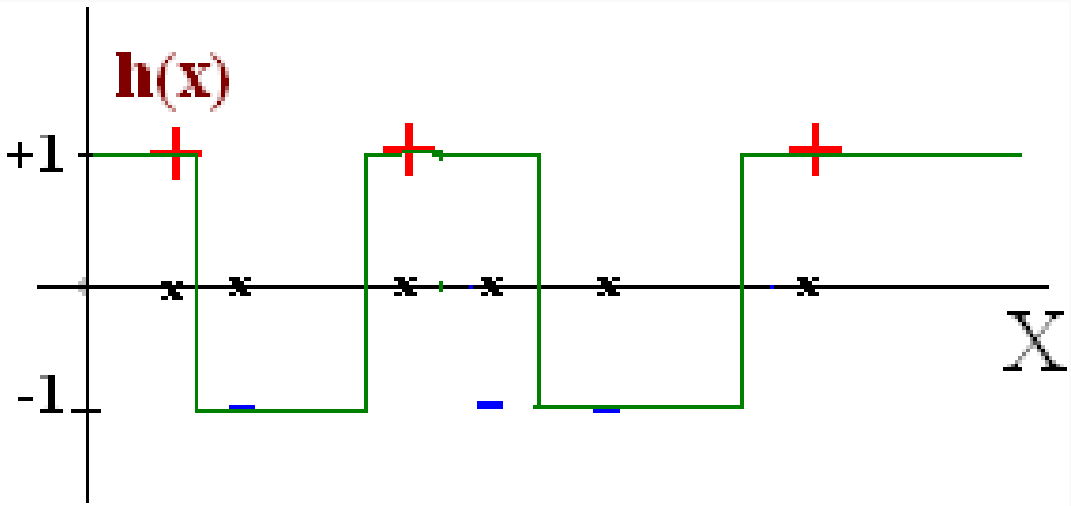
- probability that *hypothesis* h makes a mistake in predicting the label y from x, where (x, y) of Z drawn according to P

Empirical error

- For any binary function h, the *empirical error* of h based on a sample ζ is

$$L_m(h) = \frac{1}{m} \sum_{i=1}^{m} \mathbb{I}\{h(x_i) \neq y_i\}$$

- It is proportion of examples (x_i, y_i) in ζ on which h is wrong



Learning algorithm

- P is fixed (unknown)
- H is fixed *hypothesis class* of classifiers, e.g., binary functions on X (H may be infinite)
- Learn directly over H
- Let *optimal error* be $L^* = \inf_{h \in H} L(h)$
- $A : \bigcup_{m=1}^{\infty} Z^m \to H$ is a *classification learning algorithm* for H if there exists a function $\epsilon(m, \delta)$ such that for all $m \geq 1$, $0 < \delta < 1$, P, with probability at least $1 - \delta$ over $\zeta \in Z^m$ drawn according to P^m,

$$L(A(\zeta)) \leq L^* + \epsilon(m, \delta)$$

  and for all $0 < \delta < 1$, $\epsilon(m, \delta) \to 0$ with increasing m
- Probably Approximately Correct (PAC) framework* (ϵ accuracy, δ confidence parameters)

*Haussler (1992), Valiant (1984)

Empirical-error minimization

Empirical-error minimization (ERM))[†] *algorithm* A:

$$L_m(A(\zeta)) = \min_{h \in H} L_m(h)$$

[†] Vapnik (1998)

Is *empirical-error minimization a* learning *algorithm ?*

Standard generalization bounds

- Yes, if H is finite,
$$\epsilon(m, \delta) = \frac{1}{m} \ln \left(\frac{|H|}{\delta} \right)$$
- Yes, if H has a finite VC-dimension d,
$$\epsilon(m, \delta) = \sqrt{\frac{32}{m} \left(d \ln \left(\frac{2em}{d} \right) + \ln \left(\frac{4}{\delta} \right) \right)}$$
provided $m \geq d/2$.
- Yes, if H has a finite VC-dimension d *and* $L^* = 0$ then
$$\epsilon(m, \delta) = \frac{2}{m} \left(d \ln \left(\frac{2em}{d} \right) + \ln \left(\frac{2}{\delta} \right) \right)$$

VC-dimension

- *Vapnik-Chervonenkis (VC) dimension* (trace number in combinatorics)

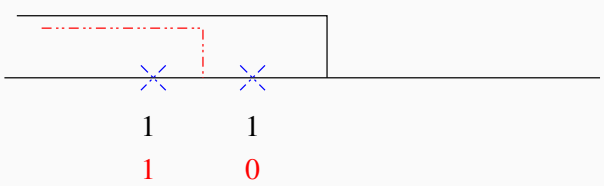
$$VC(H) = \max\{|S| : S \subset X, \mathrm{tr}_H(S) = \mathcal{P}(S)\}$$

where $\mathcal{P}(S)$ is the power set of S

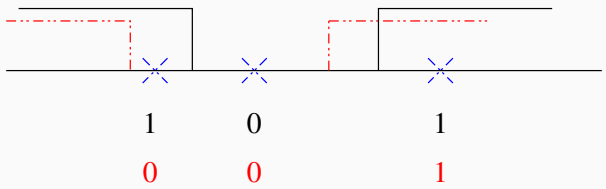
`Example`: $X = \mathbb{R}$, let $L_z = \{x : x \leq z\}$, $R_z = \{x : x > z\}$, $z \in X$

(a) $\mathcal{H} = \{h : A_h = L_z, z \in \mathbb{R}\}$, $VC(\mathcal{H}) = 1$

(b) $\mathcal{H} = \{h : A_h = L_z \bigcup R_y, y, z \in \mathbb{R}\}$, $VC(\mathcal{H}) = 2$



(a)



(b)

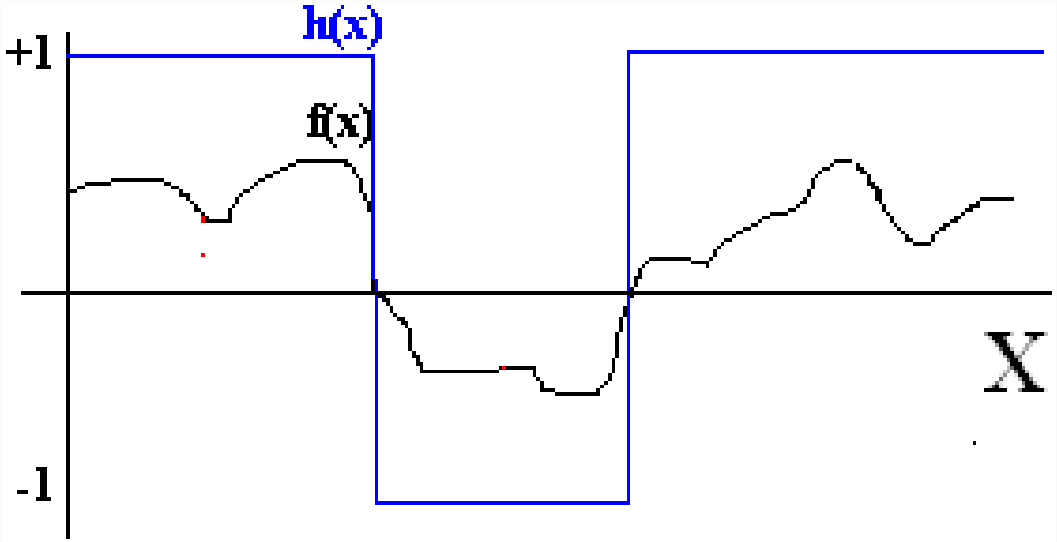
- Generally, VC-dimension can also be infinite

Learning classification indirectly via real-valued functions

Most machine learning algorithms rely on real-valued functions f (models) for classification, e.g., Neural Networks, SVM, density-estimators (e.g., Bayesian models) do classification via Max a Posteriori, etc.

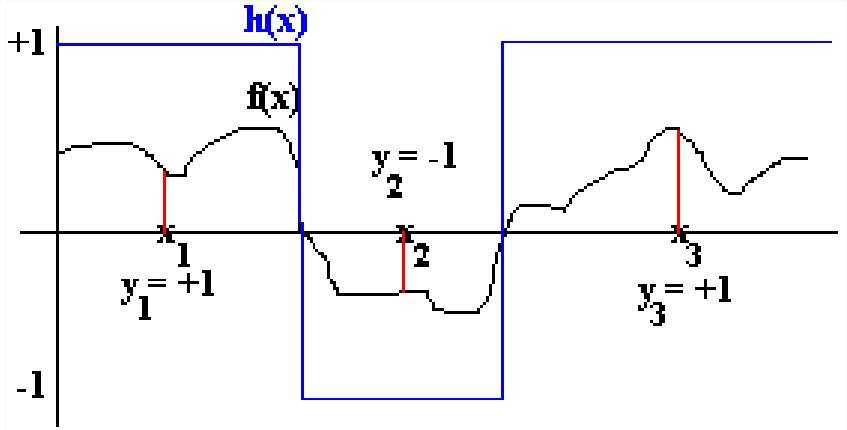
A classifier is obtained by quantizing the real value to a finite set of classes, e.g., binary classification by $\text{sgn}(f)$

- let F denote a class of real-valued functions $f : X \to \mathbb{R}$
- Can define a classifer h by $h(x) = \text{sgn}\,(f(x))$



Margin

- Typically, learning algorithms (e.g., neural networks) tend to produce classifiers having a large margin on most of the examples
- *Margin* of f on (x, y) is defined as $\mu_f(x, y) = yf(x)$



- The *sample-margin* is $\mu_\zeta(f) = \min_{\{(x,y)\in\zeta\}} \mu_f(x, y)$

Margin error of f

- For any real function f, the margin error of f is defined as

$$\begin{aligned} L^{(\gamma)}(f) &= P(\mu_f(x,y) < \gamma) \\ &= P(y\, f(x) < \gamma) \end{aligned}$$

- probability that f has γ margin error in predicting y from x

Empirical margin error

- For any $\gamma > 0$, $\zeta \in Z^m$, the *margin error* of f based on a ζ is

$$L_m^{(\gamma)}(f) = \frac{1}{m} \sum_{i=1}^{m} \mathbb{I}\{\mu_f(x_i, y_i) < \gamma\}$$

- It is proportion of examples (x_i, y_i) in ζ on which f has margin error

Learning classification with real functions

- P is fixed (unknown)
- F is fixed *hypothesis class* of real functions on X (F may be infinite)
- optimal error at 'level' γ, $L^{*(\gamma)} = \inf_{f \in F} L^{(\gamma)}(f)$,
- $A : \bigcup_{m=1}^{\infty} Z^m \times \mathbb{R}_+ \to F$ is a *classification learning algorithm* for F if there exists a function $\epsilon(m, \gamma, \delta)$ such that for all $m \geq 1$, $0 < \delta < 1$, P, and $\gamma > 0$ with probability at least $1 - \delta$ over $\zeta \in Z^m$ drawn according to P^m,

$$L(A(\zeta, \gamma)) \leq L^{*(\gamma)} + \epsilon(m, \gamma, \delta)$$

(as before, for all $0 < \delta < 1$, $\gamma > 0$, $\epsilon(m, \gamma, \delta) \to 0$ with increasing m)

$L^{*(\gamma)}$ can be higher than L^* so why is it interesting?

Advantage of large-margin

PAC learning is about ensuring that the following 'Bad Event' has *low* probability δ (namely, high confidence $1-\delta$):

- Direct classification: infinite hypothesis class H of discrete functions h

  Bad Event: for some $h \in H$, the empirical error $L_m(h)$ is not an accurate estimate of generalization error $L(h)$

- Indirect classification: infinite class F of real functions

  Bad Event: for some $f \in F$, the empirical error $L_m^{(\gamma)}(f)$ is not an accurate estimate of the generalization error $L(f)$

  Technical Trick show that this bad event implies a new bad event involving finite class $\hat{F}_\gamma$

New Bad Event: for some hypothesis $\hat{f} \in \hat{F}_\gamma$, $L_m^{(\gamma)}(\hat{f})$ is not an accurate estimate of $L^{(\gamma)}(\hat{f})$

Why is this an advantage?

- cardinality $\left|\hat{F}_\gamma\right|$ decreases as γ increases
- δ is linearly proportional to cardinality $\left|\hat{F}_\gamma\right|$
- therefore as margin γ increases $\Rightarrow$ confidence $1-\delta$ increases
- alternatively, larger margin $\gamma \Rightarrow$ smaller sample size m (for a given confidence δ)

Benefit of Large Margin

If an algorithm aims at finding f such that margin-error $L_m^{(\gamma)}(f)$ is small with large γ,

Then its generalization error $L(f)$ will be small even if the sample size m is small

Useful for Machine Learning

- In the late 90's, the success of learning linear half spaces* (SVM) in high dimensional input spaces and Neural Networks, was explained† by this principle
- In work‡ on Neural Networks, it is shown:

  generalization error bound

  $$L_m^{(\gamma)} + \sqrt{\frac{1}{m}\left(\left(\frac{W}{\gamma}\right)^2 n \log\left(\frac{W}{\gamma}\right)\log^2 m + \frac{1}{\delta}\right)}$$, where n is input dimension, W is upper bound on sum of the weights' norm

  automatic overfitting avoidance weights may be allowed to grow as long as the margin γ is also large (gradient descent algorithm tends to increase weights to reduce the empirical error)

  Regularization essential in training large generative-AI models

*Cristianini and Shawe-Taylor (2000), Vapnik (1998)

†Anthony and Bartlett (1999)

‡Bartlett (1998)

Motivation for learning with large width

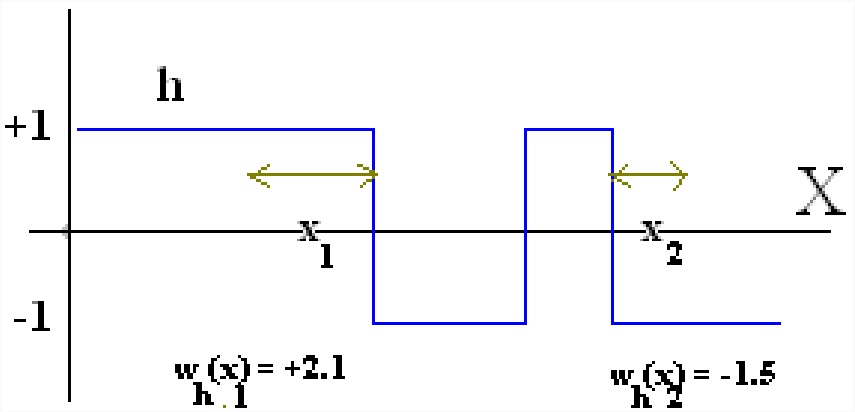
- Could a class H of discrete functions h be learned directly (not indirectly via real-valued functions) and still take advantage of large margin?
- Consider binary classifiers (mapping to $\{-1, 1\}$)

  Margin is not useful because the distance from zero is always ± 1
  New notion: Width

Width

- For $x \in X$, define the *width* of h on x by

$$w_h(x) = \sup\{a \geq 0 : h(z) = h(x),\ x - a \leq z \leq x + a\}.$$

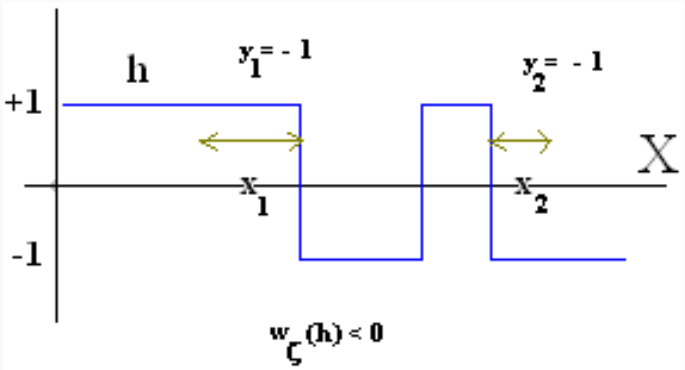


- define the *signed width* of h on x by

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

Sample width

- Let $Z = X \times Y$
- A finite *sample* $\zeta = \{(x_i, y_i)\}_{i=1}^{m}$ is an element of Z^m
- y_i is label of i^{th} example
- *Sample width* of h, $\mathrm{w}_\zeta(h) = \min_{(x,y)\in\zeta} y\, f_h(x)$.

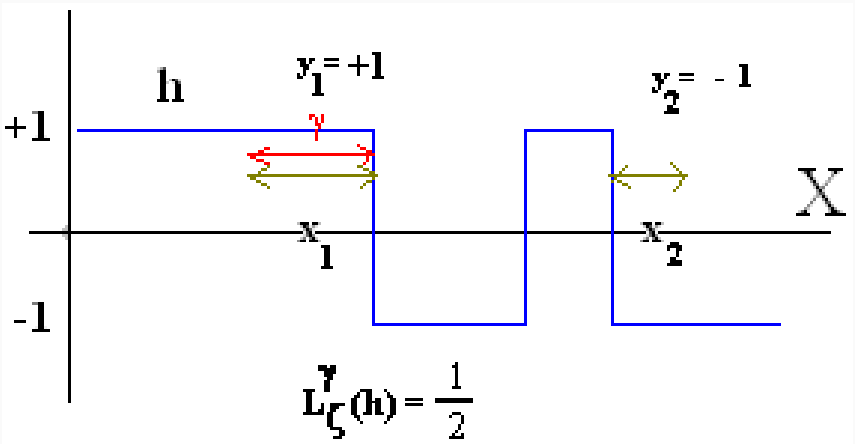


- So if $\mathrm{w}_\zeta(h) = \gamma > 0$, then for each (x, y) in the sample, $h(x) = y$ and is constant on an interval of the form $\langle x - \gamma, x + \gamma \rangle$

Width Error

- For $\gamma > 0$ and a sample $\zeta \in Z^m$, width-error, or γ-*error* of a binary function h is

$$L_m^{(\gamma)}(h) = \frac{1}{m} \sum_{i=1}^{m} \mathbb{I}\{y_i f_h(x_i) < \gamma\}$$



Learning binary functions on the real line

Theorem 1

Let $B > 0$ and denote the domain by $X = [0, B]$ with range $Y = \{-1, +1\}$ and let $Z = X \times Y$. Let P be a probability distribution on Z and suppose that $\delta \in (0, 1)$. Then, with P^m-probability at least $1 - \delta$, $\zeta \in Z^m$ is such that for any *function $h : X \to Y$ and for* all *$\gamma > 0$,*

$$L(h) < L_m^{(\gamma)}(h) + \epsilon(m, \gamma, \delta),$$

where

$$\epsilon(m, \gamma, \delta) = \sqrt{\frac{8}{m}\left(\frac{2B}{\gamma}\ln 3 + \ln\left(\frac{32B}{\delta\gamma}\right)\right)}.$$

*Anthony, M., & Ratsaby, J. (2010). Maximal width learning of binary functions. *Theoretical Computer Science*, *411*(1), 138–147

Width improves the margin-based generalization bounds

- γ is not prescribed in advance.
- If we used indirect classification (via real-valued functions f) then ϵ is of order $\sqrt{d_\gamma (\ln m)^2 / m}$ where d_γ is the fat-shattering dimension (scale-sensitive analogue of the VC dimension that applies to real-valued function classes)
- d_γ is of order $1/\gamma$, so margin-based bounds (not width-based) involve a term of order $\sqrt{(\ln m)^2/(\gamma m)}$ (worse than the width-based bound of Theorem 1)
- Similar improvement holds for the restricted case ($L^* = 0$)

Learning classification with large width–in general

So far we considered large-width learning of binary classification on the real line.

We can generalize to other input spaces X with a distance d (or metric), and $\text{dist}(x, S) := \min_{x' \in S} d(x, x')$ (inf in the case S is infinite)

Sets S_+, S_- in X, points labeled 1 and -1, respectively

Signed width

- $f_h(x) := \frac{1}{2}(\text{dist}(x, S_-) - \text{dist}(x, S_+))$, or
- $f_h(x) := h(x)\text{dist}\left(x, S_{\overline{h(x)}}\right)$

Other input spaces

multi-dimensional cube[*] binary classification over $[0,1]^n$, S_- and S_+ are *union of boxes* labeled -1 and 1

multi-dimensional cube[†] multi-category classification on $[0,1]^n$ where, for category l, the points with classification l form a set S_l that is a union of boxes

general finite metric space[‡] all binary functions on any finite metric space, where S_+ and S_- form any partition of the space

Infinite totally bounded metric space[§] multi-category classification on any metric space X for which there exists a finite cover

---

[*] Anthony, M., & Ratsaby, J. (2014a). A hybrid classifier based on boxes and nearest neighbors. *Discrete Applied Mathematics*, *172*, 1–11

[†] Anthony, M., & Ratsaby, J. (2012). Analysis of a multi-category classifier. *Discrete Applied Mathematics*, *160*(16-17), 2329–2338

[‡] Anthony, M., & Ratsaby, J. (2014b). Learning bounds via sample width for classifiers on finite metric spaces. *Theoretical Computer Science*, *529*, 2–10

[§] Anthony, M., & Ratsaby, J. (2016). Multi-category classifiers and sample width. *Journal of Computer Systems and Sciences*, *82*(8), 1223–1231

general *finite* distance space (*no need* to satisfy the metric properties, except non-negativity)

- learn binary classification by half-spaces* defined by pairs $[p^+, 1]$, $[p^-, -1] \in X \times \{-1, 1\}$, where $h(x) = 1$ if $d(x, p^+) < d(x, p^-)$, and signed width $f_h(x) := h(x)\left(\max\{d(x, p^+), d(x, p^-)\} - \min\{d(x, p^+), d(x, p^-)\}\right)$
- learn binary classification by nearest-prototype† classifiers , where S_+ and S_- are set of points labeled 1 and -1, respectively, and may include a subset of the sample ζ (hence, this includes nearest-neighbor classifiers)

general possibly *infinite* distance space‡ learn binary classification by half-spaces (as in *)

---

*Anthony, M., & Ratsaby, J. (2018b). Large-width bounds for learning half–spaces on distance spaces. *Discrete Applied Mathematics*, *243*, 73–89

†Anthony, M., & Ratsaby, J. (2018a). Large width nearest prototype classification on general distance spaces. *Theoretical Computer Science*, *738*, 65–79

‡Ratsaby, J. (2023). Learning half-spaces on general infinite spaces equipped with a distance function. *Information and Computation*, *291*, 105008

general problem-solution space* similarity function on space of problems, and on the space of solutions, learn by case-based inference (CBI)

*Anthony, M., & Ratsaby, J. (2015). A probabilistic approach to case-based inference. *Theoretical Computer Science*, *589*, 61–75

Large Width learning algorithm

Algorithm LW* for learning classification with large width on Euclidean space, is implemented and distributed in the WEKA machine learning platform†

Available via WEKA package manager‡

*Anthony, M., & Ratsaby, J. (2020). Large-width machine learning algorithm. *Progress in Artificial Intelligence*, *9*(3), 275–285

†Frank et al. (2016)

‡Anthony, M., & Ratsaby, J. (2025). *Algorithm LW for WEKA*.

Thanks for your attention.