

CMT311 Principles of Machine Learning

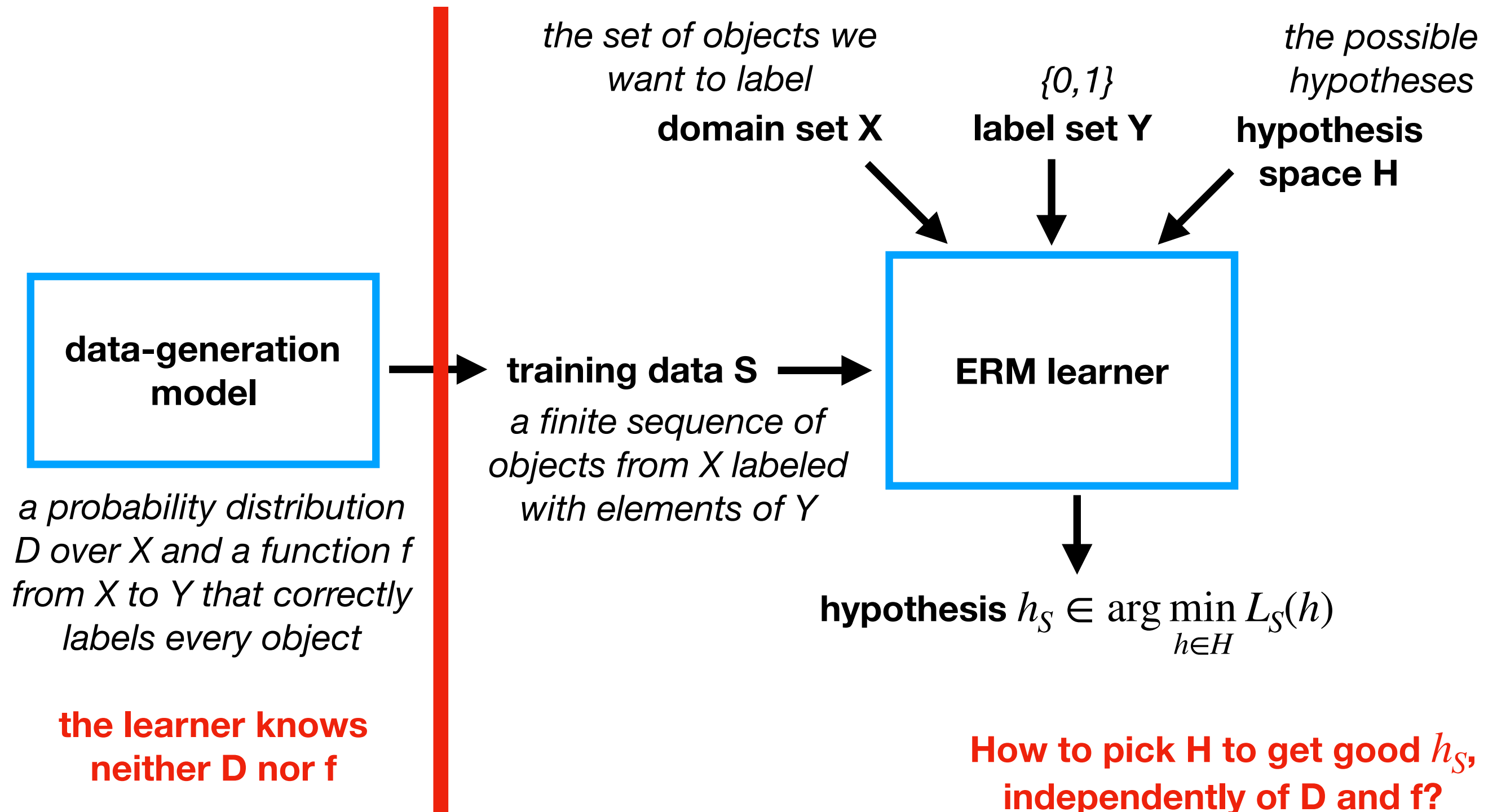
ERM & PAC Learning

Angelika Kimmig
KimmigA@cardiff.ac.uk

18.10.2019

ERM Learning

(Boolean functions)



Empirical Risk Minimisation (ERM)

- The **training error** (also called **empirical error** or **empirical risk**) of hypothesis h with respect to training sample $S = ((x_1, y_1), \dots, (x_m, y_m))$ is the **fraction** of the training sample h is **not consistent** with, i.e.,

$$L_S(h) = \frac{\left| \{i \in \{1, \dots, m\} \mid h(x_i) \neq y_i\} \right|}{m}$$

- The learner can compute this for any given hypothesis!
- An **ERM (empirical risk minimisation) learner** returns a hypothesis h that minimises $L_S(h)$ given S

ERM Learning

- Under the following conditions, ERM will not overfit:

- H is finite

not a necessary condition (more later)

- there is a $h \in H$ such that $L_{D,f}(h) = 0$

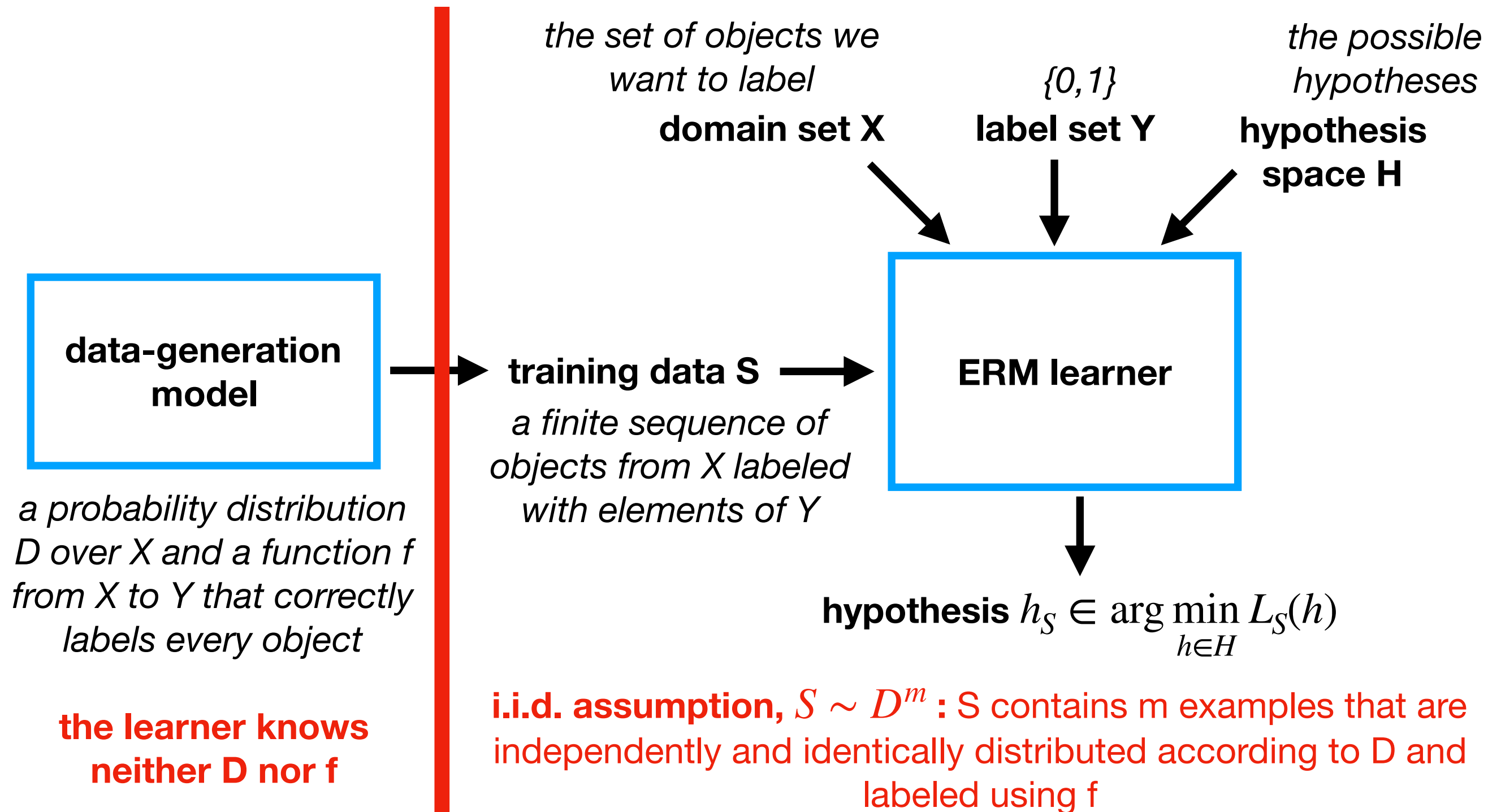
the **realisability** assumption

note: realisability
implies $L_S(h_S) = 0$

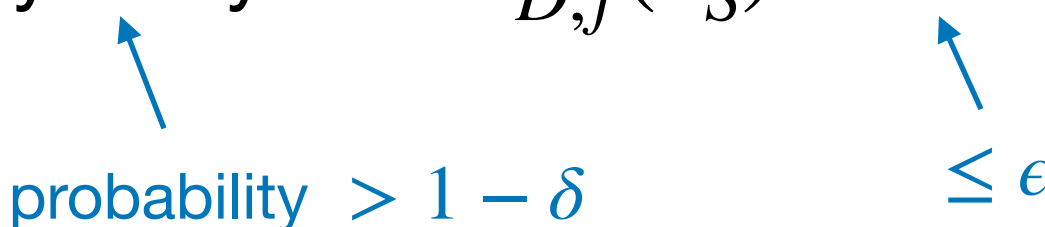
- S is “large enough”

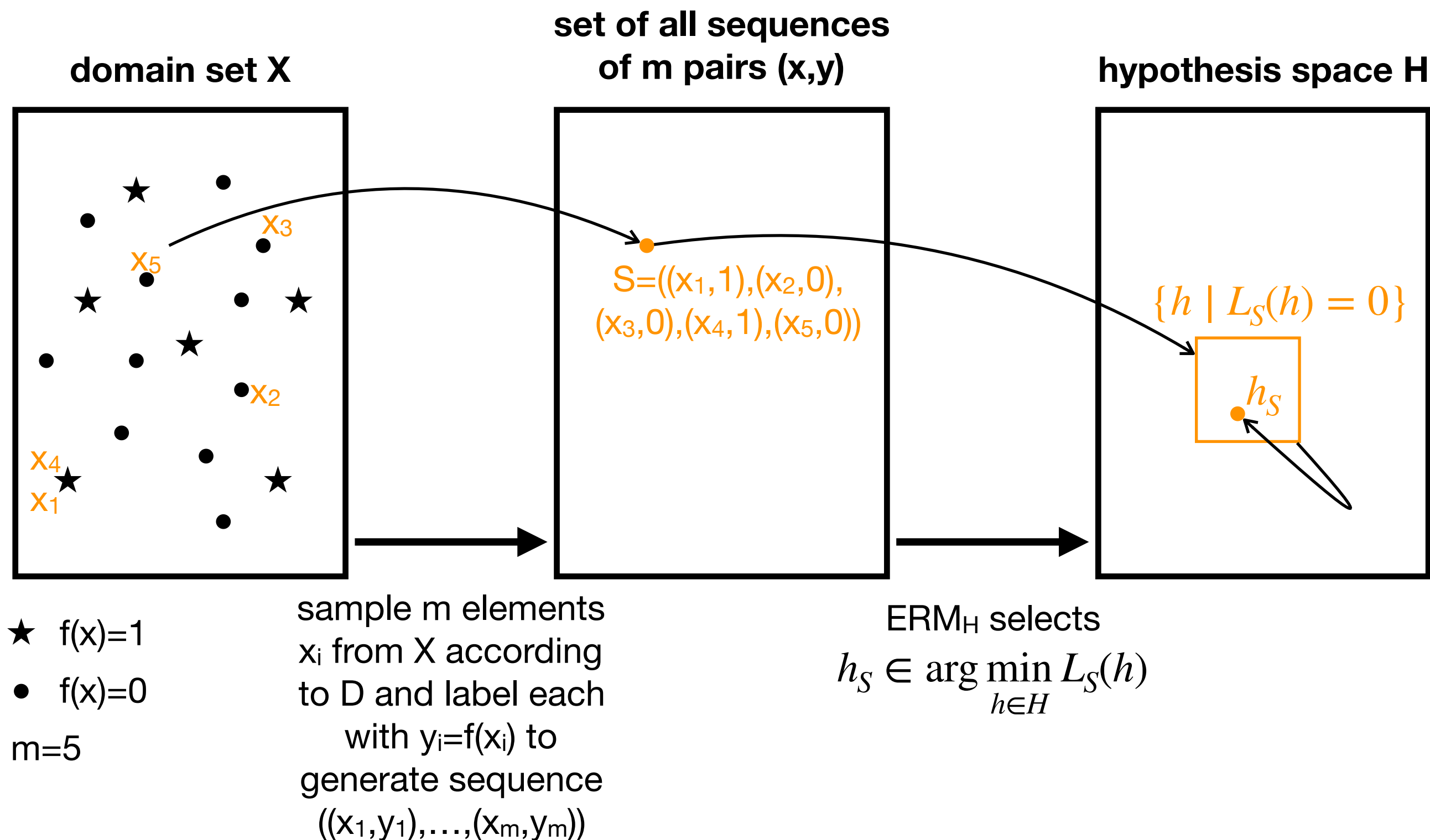
we’ll make this precise next

ERM Learning

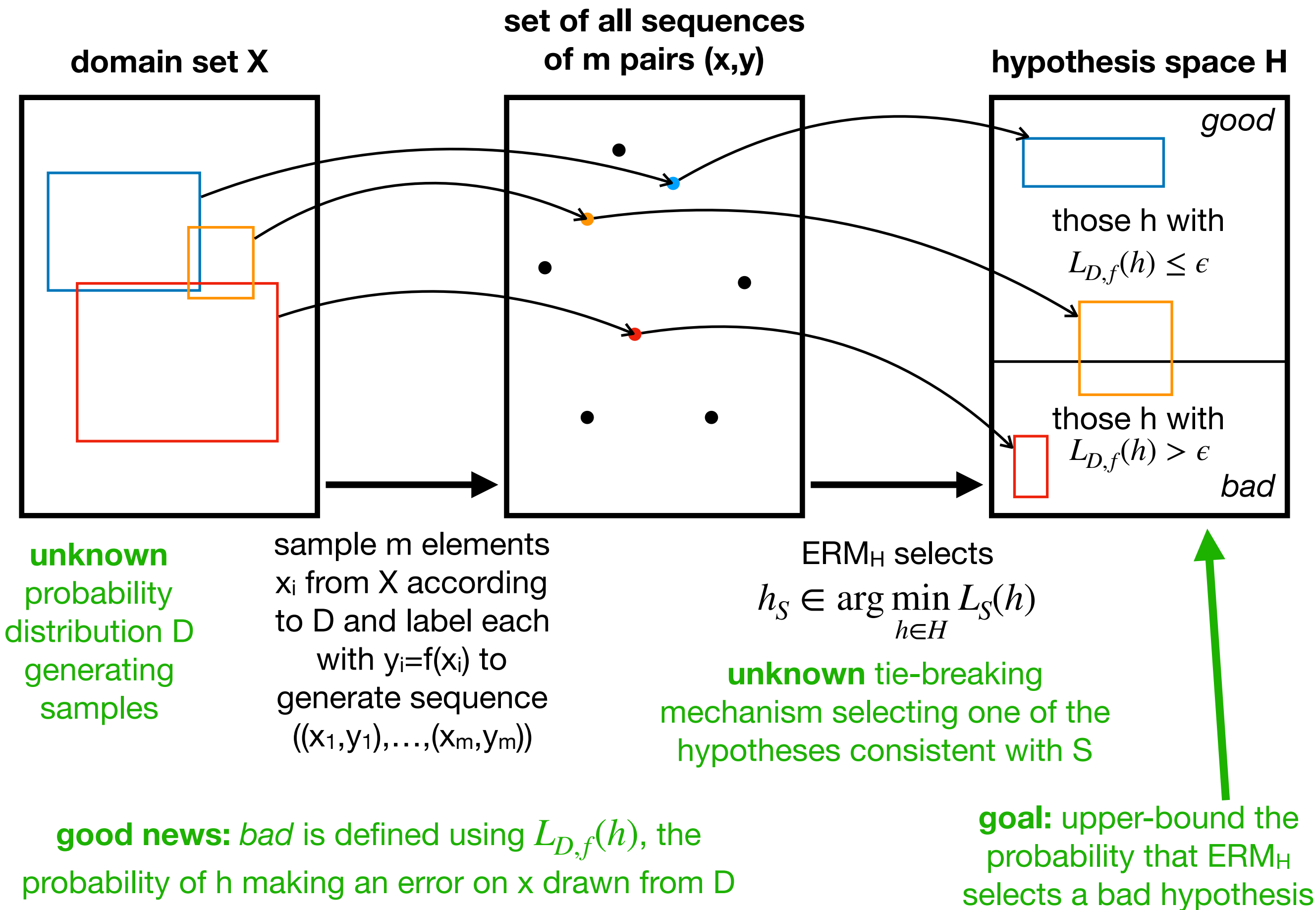


ERM Learning

- ideally, we'd want ERM to return h_S with $L_{D,f}(h_S) = 0$
- this is not realistic: the random process may give us a misleading S
- instead, we aim for h_S that is **probably approximately correct**, i.e., for which it is very likely that $L_{D,f}(h_S)$ is small for a randomly selected S




what is the probability that the true error of h_S is small?



Formally

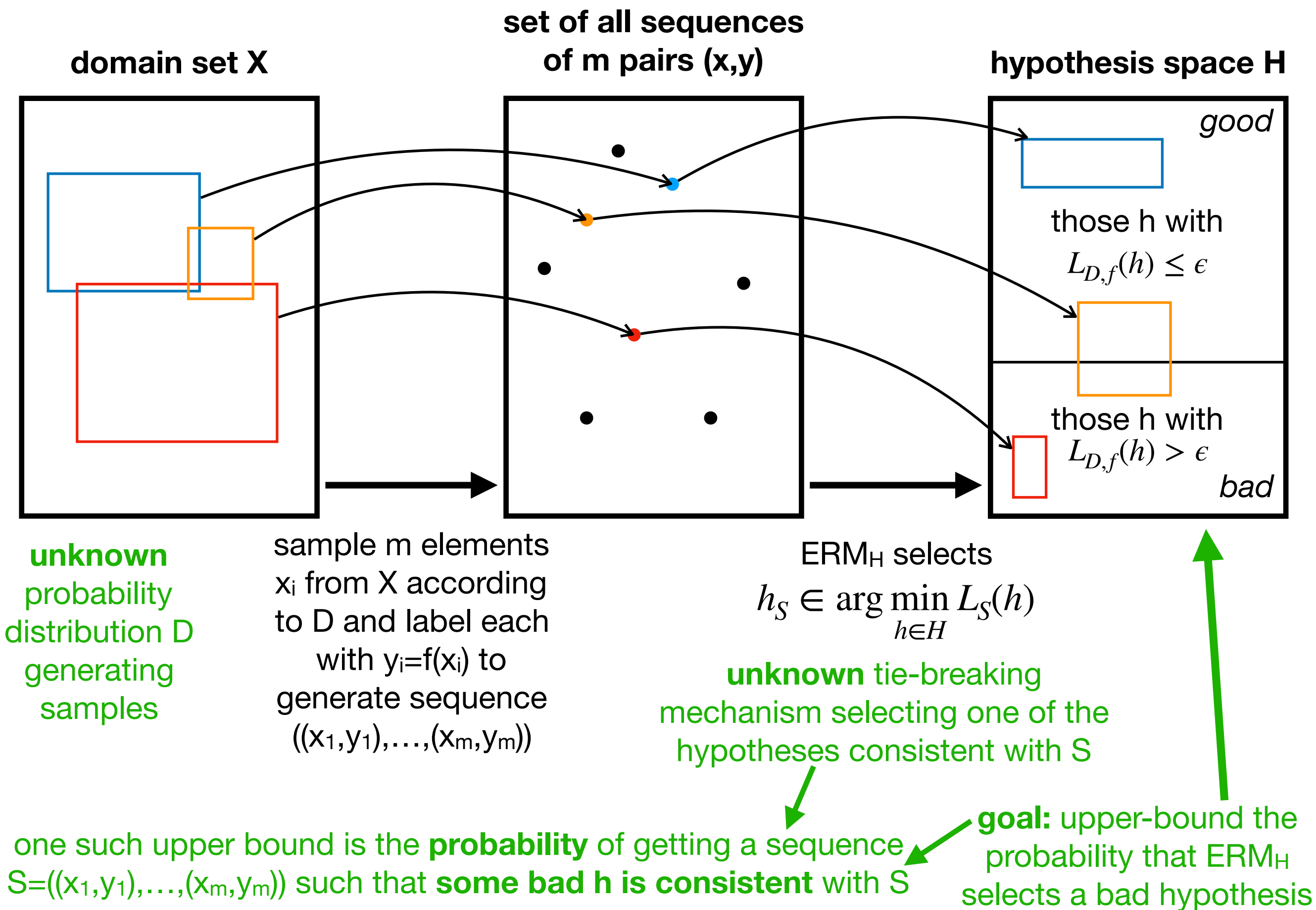
- Fix an **accuracy parameter** ϵ , and consider $L_{D,f}(h_S) > \epsilon$ a **failure** of the learner.
- **Goal:** ensure that the probability of failure (over samples S drawn from D and labeled by f) is at most δ , where we call $(1 - \delta)$ the **confidence parameter**.
- That is, given parameters ϵ and δ , we want $P(L_{D,f}(h_S) > \epsilon) \leq \delta$ or equivalently $P(L_{D,f}(h_S) \leq \epsilon) > 1 - \delta$
- Question: how large should S be for this to hold?

Basic process

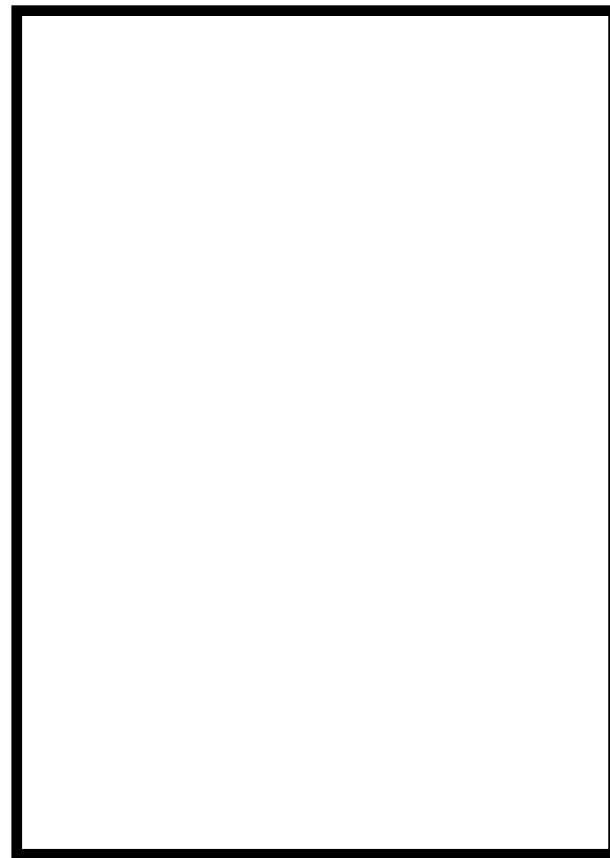
- The learner knows the object set X and hypothesis space H .
- The learner chooses the parameters ϵ and δ .
- The learner does not know the distribution D and function f , but can request an arbitrary but fixed number m of training examples drawn i.i.d. from D and labeled using f .
- How many examples should the learner ask for to achieve $P(L_{D,f}(h_S) > \epsilon) \leq \delta$?

Which m to choose?

- How many examples should the learner ask for to achieve $P(L_{D,f}(h_S) > \epsilon) \leq \delta$?
- We'll answer this question by
 - providing a function $g(m)$ such that $P(L_{D,f}(h_S) > \epsilon) \leq g(m)$
preview: $g(m) = |H| e^{-\epsilon m}$
 - rearranging $g(m) \leq \delta$ to obtain an inequality with just m on one side
preview: $m \geq \frac{\log(|H|/\delta)}{\epsilon}$

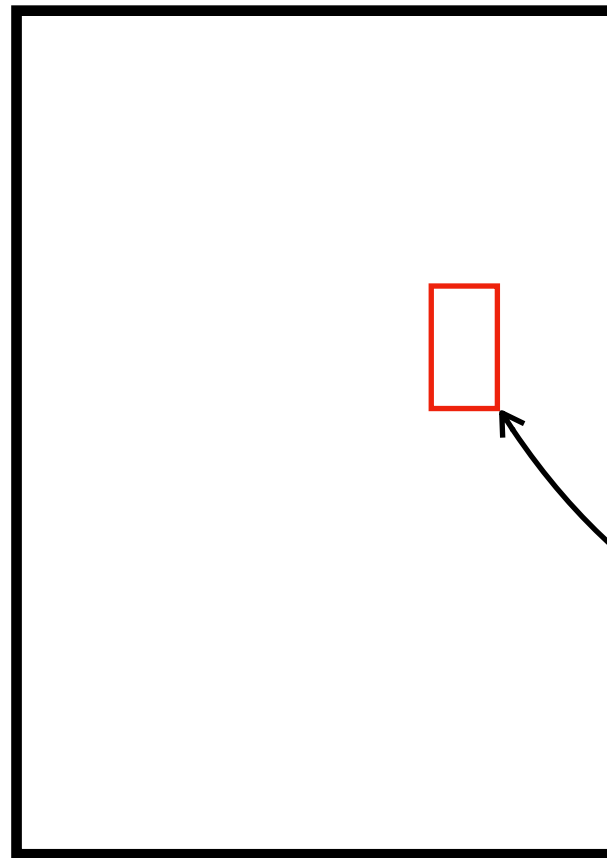


domain set X



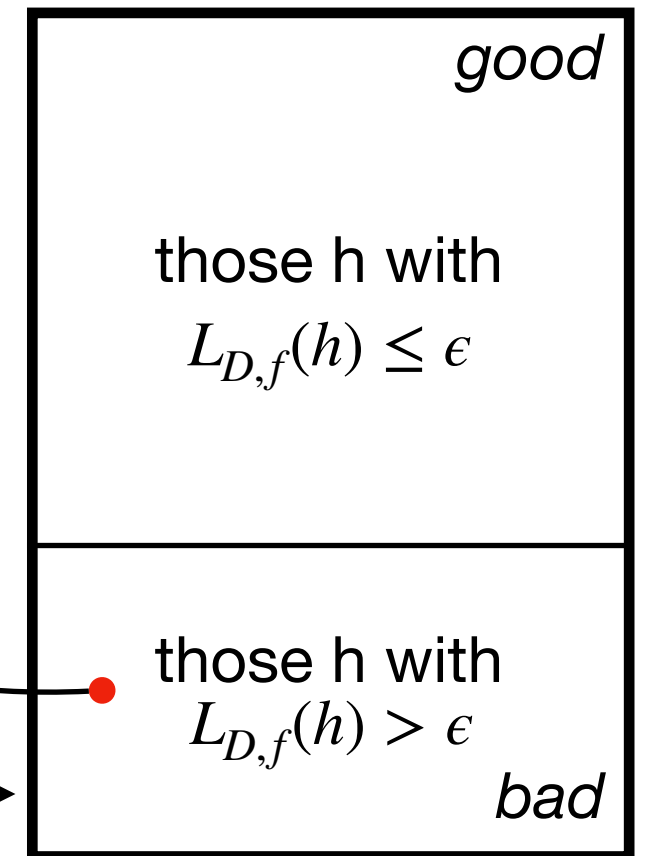
sample m elements x_i from X
according to D and label each
with $y_i=f(x_i)$ to generate
sequence $((x_1, y_1), \dots, (x_m, y_m))$

set of all sequences
of m pairs (x, y)



ERM_H selects
$$h_S \in \arg \min_{h \in H} L_S(h)$$

hypothesis space H



for a specific bad hypothesis h , what is the probability of getting a sequence
 $S=((x_1, y_1), \dots, (x_m, y_m))$ such that this h is consistent with S ?

for a specific bad hypothesis h , what is the probability of getting a sequence $S=((x_1,y_1),\dots,(x_m,y_m))$ such that this h is consistent with S ?

↑
for each x_i , $h(x_i)=y_i$

↑
for each x_i , $h(x_i)=f(x_i)$

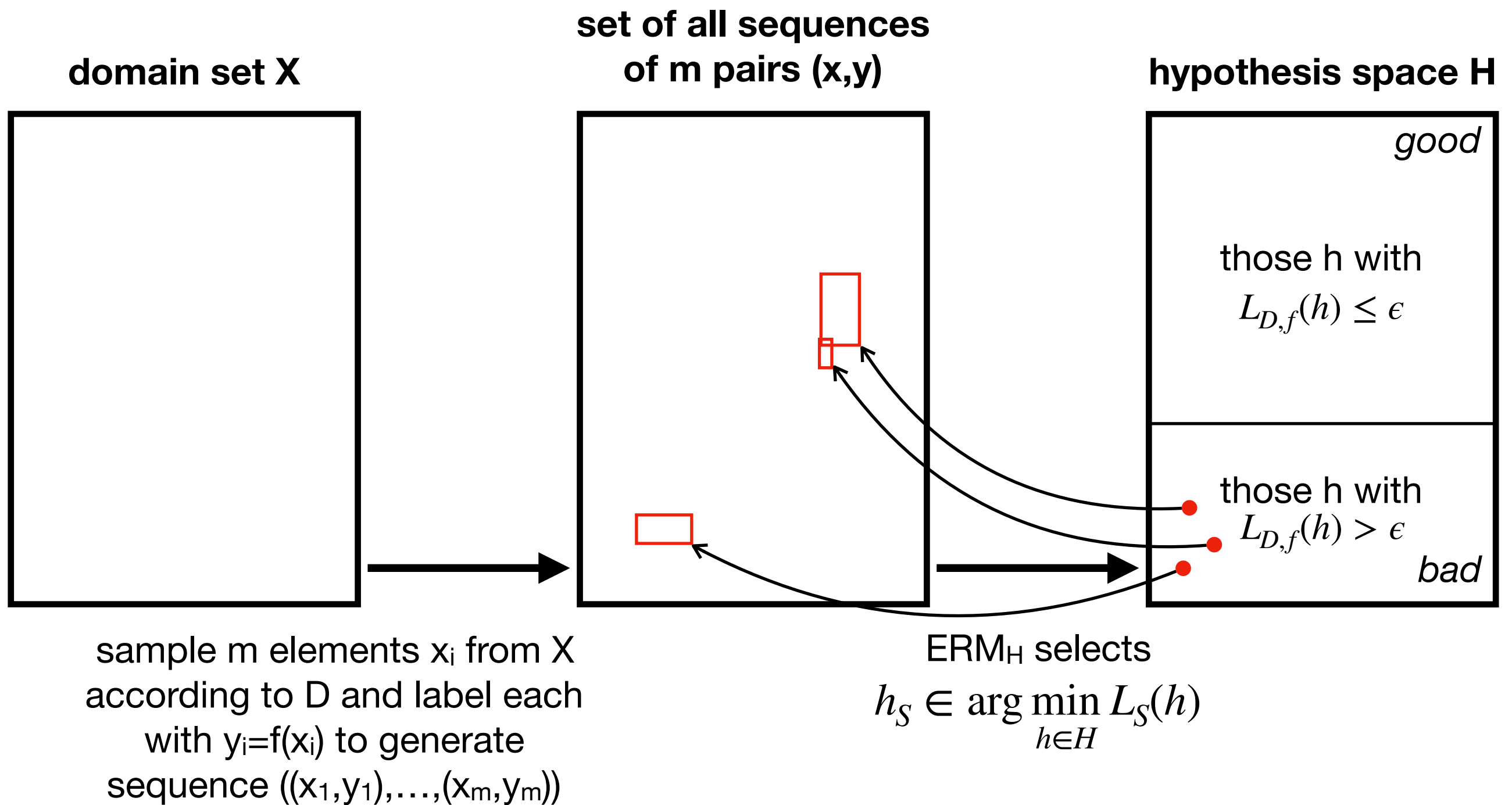
recall that $L_{D,f}(h)$ is the probability that for x drawn from D , $h(x) \neq f(x)$

thus, $1 - L_{D,f}(h)$ is the probability that for x drawn from D , $h(x) = f(x)$

as each x_i in S is drawn i.i.d. from D ,

the **probability of getting S consistent with h** is $(1 - L_{D,f}(h))^m \leq (1 - \epsilon)^m$

↖ h is bad



for a specific bad hypothesis h , what is the probability of getting a sequence $S=((x_1,y_1), \dots, (x_m,y_m))$ such that this h is consistent with S ? $\leq (1 - \epsilon)^m$

the probability of getting S consistent with some bad h is $\leq |H_{bad}| (1 - \epsilon)^m$

$\leq |H| (1 - \epsilon)^m \leq |H| e^{-\epsilon m}$

holds for all $\epsilon \in [0,1]$

Which m to choose?

- How many examples should the learner ask for to achieve $P(L_{D,f}(h_S) > \epsilon) \leq \delta$?
- We'll answer this question by
 - providing a function $g(m)$ such that $P(L_{D,f}(h_S) > \epsilon) \leq g(m)$
preview: $g(m) = |H| e^{-\epsilon m}$
 - rearranging $g(m) \leq \delta$ to obtain an inequality with just m on one side
preview: $m \geq \frac{\log(|H|/\delta)}{\epsilon}$

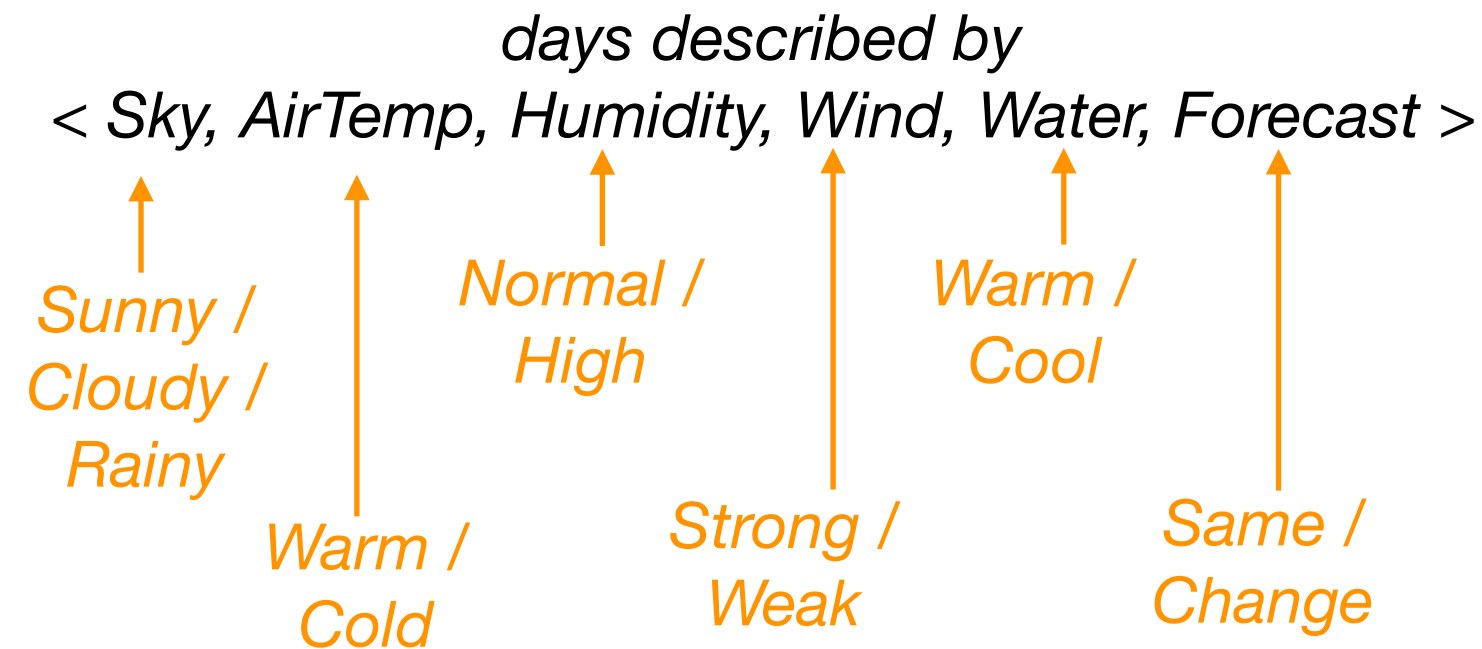
ERM Learning

Let H be a finite hypothesis space of Boolean functions on X , $\delta \in [0,1]$, $\epsilon \in [0,1]$, and m an integer satisfying $m \geq \frac{\log(|H|/\delta)}{\epsilon}$. Then, for any distribution D over X and any labeling function f for which the realisability assumption holds, with probability of at least $1 - \delta$ over the choice of an i.i.d. sample S of size m , we have that for every ERM hypothesis h_S it holds that $L_{D,f}(h_S) \leq \epsilon$.

*That is, for sufficiently large m , any ERM hypothesis is **probably** (with confidence $1 - \delta$) **approximately** (up to an error of ϵ) **correct**.*

Example

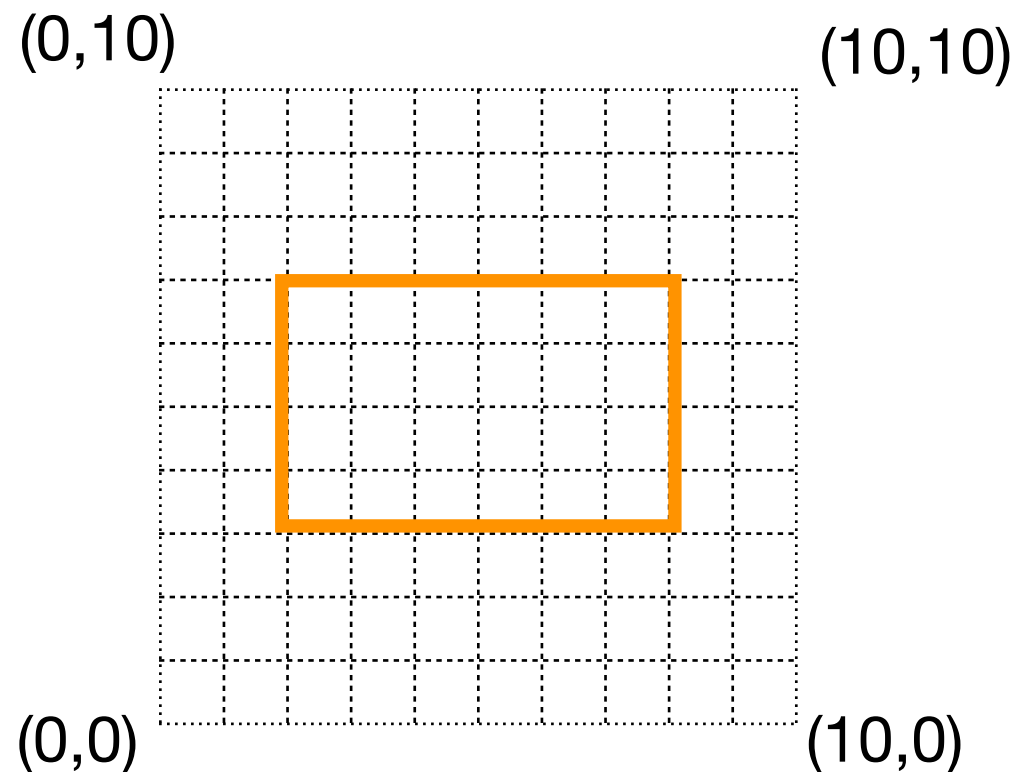
$$m \geq \frac{\log(|H|/\delta)}{\epsilon}$$



$$|H| = 1 + 4 \cdot 3 \cdot 3 \cdot 3 \cdot 3 \cdot 3 = 973$$

ϵ		δ	
		0.05	0.01
0.05	$m \geq \frac{\log(973/0.05)}{0.05} = 197.5$	$m \geq \frac{\log(973/0.05)}{0.01} = 987.6$	
0.01	$m \geq \frac{\log(973/0.01)}{0.05} = 229.7$	$m \geq \frac{\log(973/0.01)}{0.01} = 1148.6$	

Example $m \geq \frac{\log(|H|/\delta)}{\epsilon}$



$$|H| = 1 + \sum_{j=1}^{11} \sum_{i=1}^{11} ij = 4357$$

very loose bounds!
note there are only 121 points...

ϵ		δ	
		0.05	0.01
0.05	$m \geq \frac{\log(4357/0.05)}{0.05} = 227.5$	$m \geq \frac{\log(4357/0.05)}{0.01} = 1137.5$	
0.01	$m \geq \frac{\log(4357/0.01)}{0.05} = 259.7$	$m \geq \frac{\log(4357/0.01)}{0.01} = 1298.5$	

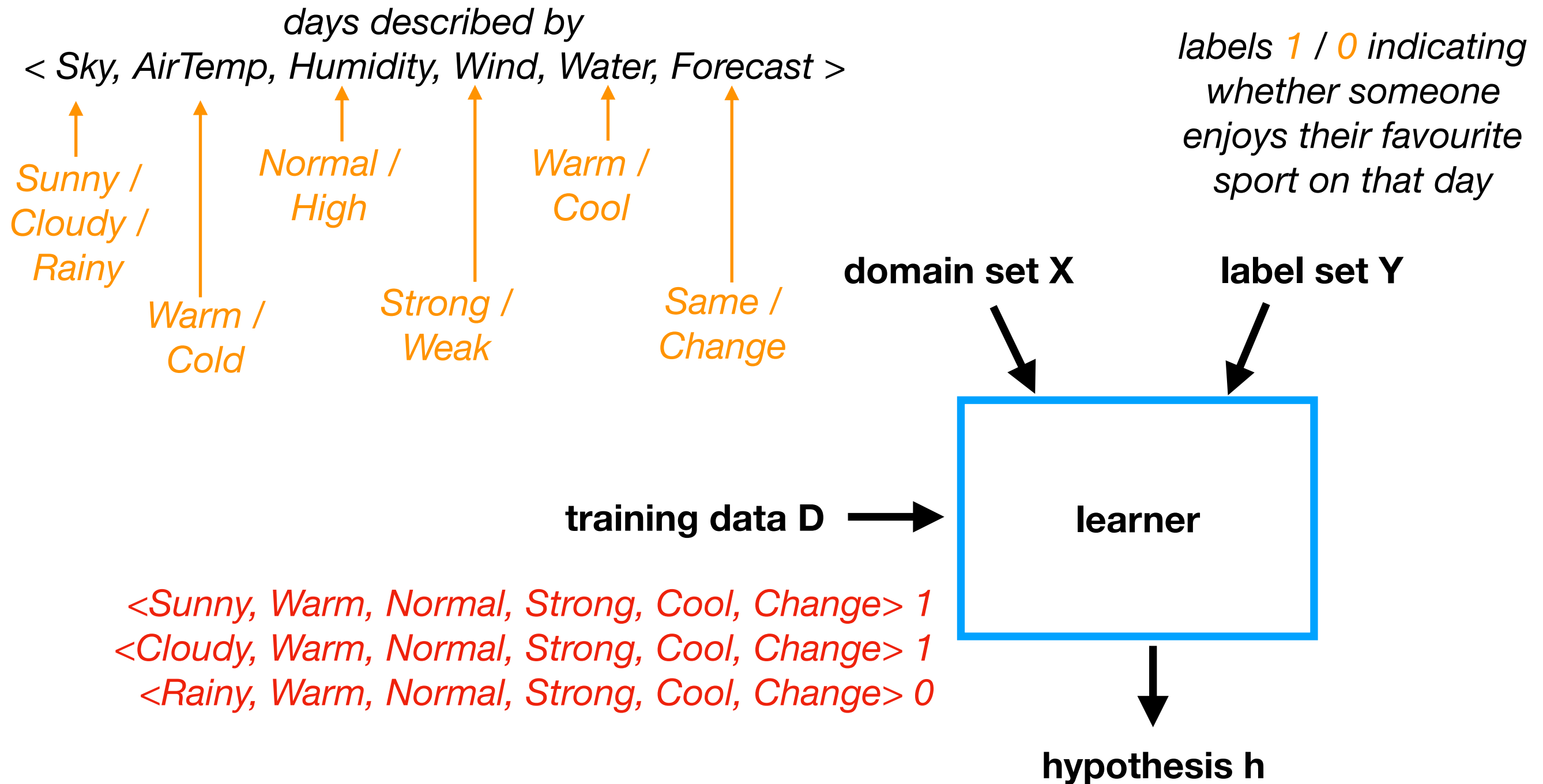
PAC Learnability

- PAC = **P**robably **A**pproximately **C**orrect
- A hypothesis class H is **PAC learnable** if there exists a function $m_H : (0,1)^2 \rightarrow \mathbb{N}$ and a learning algorithm A with the following property: For every $\epsilon, \delta \in (0,1)$, for every distribution D over X , and for every function $f : X \rightarrow \{0,1\}$, if the realisability assumption holds w.r.t. H, D, f , then if given $m \geq m_H(\epsilon, \delta)$ i.i.d. examples generated by D and labeled by f , the algorithm A returns a hypothesis h such that with probability at least $1 - \delta$ over the choice of the examples, the true error $L_{D,f}(h)$ is at most ϵ .

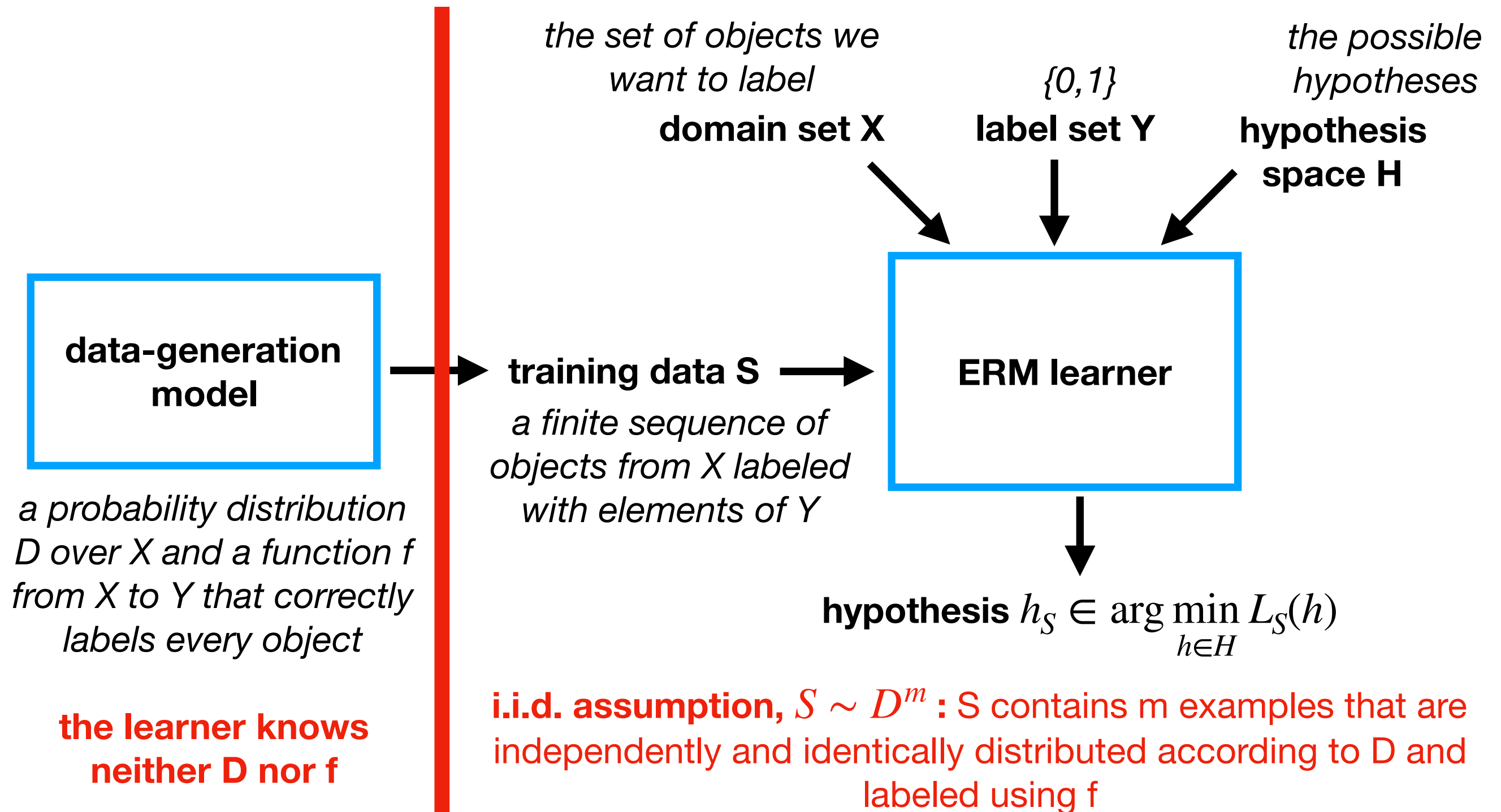
Sample Complexity

- The function $m_H : (0,1)^2 \rightarrow \mathbb{N}$ determines the **sample complexity** of learning H , i.e., the number of samples needed to guarantee a probably approximately correct solution.
- More precisely, $m_H(\epsilon, \delta)$ is the minimal integer that satisfies the requirements of PAC learning
- Thus: every finite H is PAC learnable with sample complexity
$$m_H(\epsilon, \delta) \leq \left\lceil \frac{\log(|H|/\delta)}{\epsilon} \right\rceil$$

No correct h in H

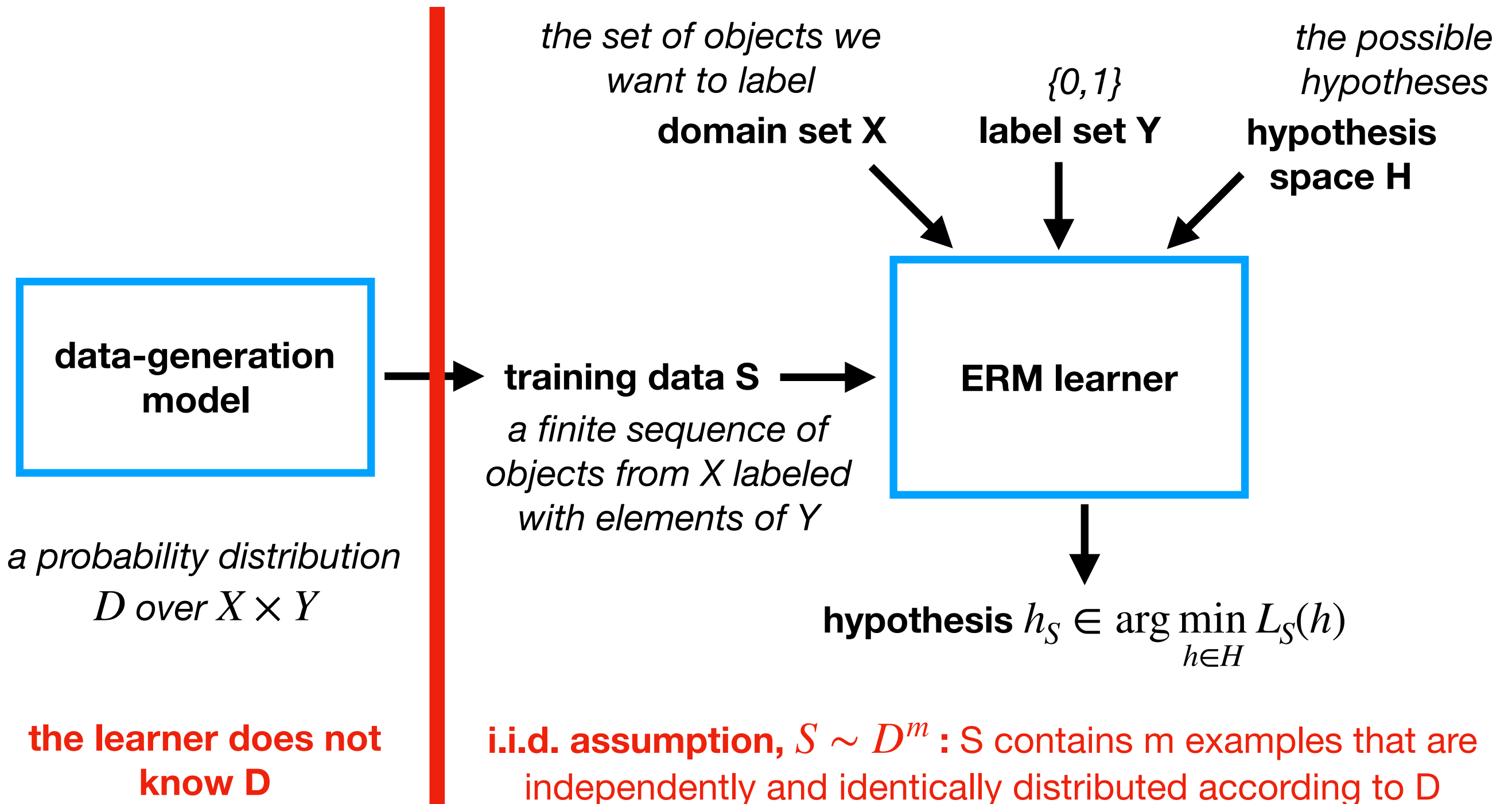


ERM Learning



ERM Learning

with randomly labeled examples



New data generation model

- We now consider a distribution D over labeled objects, e.g.,
 $D((x, y)) = D_X(x) \cdot D_Y(y \mid x)$
- Advantages:
 - can be a more realistic model of the world
 - can handle cases violating the realisability assumption
- Adapt the definition of true error to $L_D(h) = D(\{(x, y) \mid h(x) \neq y\})$
- Goal: a hypothesis that probably approximately minimises $L_D(h)$

The Bayes optimal predictor

- For any D over $X \times \{0,1\}$, the best labeling function is

$$f_D(x) = \begin{cases} 1 & \text{if } P_D(y = 1 \mid x) \geq 0.5 \\ 0 & \text{otherwise} \end{cases}$$

- best = no other $g : X \rightarrow \{0,1\}$ has lower true error
- but we do not know D ...
- instead, we'll aim to learn a predictor whose error is not much larger than the best error in a given class of predictors

Agnostic PAC Learnability

- A hypothesis class H is **agnostic PAC learnable** if there exists a function $m_H : (0,1)^2 \rightarrow \mathbb{N}$ and a learning algorithm A with the following property: For every $\epsilon, \delta \in (0,1)$, for every distribution D over $X \times Y$, if given $m \geq m_H(\epsilon, \delta)$ i.i.d. examples generated by D , the algorithm A returns a hypothesis h such that with probability at least $1 - \delta$ over the choice of the examples, the true error $L_D(h)$ is at most ϵ larger than the lowest true error of any hypothesis in H , i.e.,
$$L_D(h) \leq \min_{h' \in H} L_D(h') + \epsilon.$$

Remarks

- Agnostic PAC learnability generalises PAC learnability beyond the realisability assumption.
- Finite hypothesis classes are agnostic PAC learnable using ERM (*see the book if interested*)
- The whole setup can also be generalised beyond Boolean concept learning (*see the book if interested*)
- The original definition of PAC learnability by Valiant also imposes conditions on the time the algorithm needs to find an answer (*we'll get back to this*)
- Both PAC learning and agnostic PAC learning first fix H , and then choose an algorithm A — is there an algorithm that would work for any H ?

No-Free-Lunch Theorem

training data contains less than half of all possible elements

Let A be any learning algorithm for Boolean concept learning over domain set X . Let $m < |X|/2$. Then there exists a distribution D over $X \times \{0,1\}$ such that

there is a function with true error zero

- There exists a function $f : X \rightarrow \{0,1\}$ with $L_D(f) = 0$.
- With probability of at least $1/7$ over the choice of $S \sim D^m$ we have that $L_D(A(S)) \geq 1/8$.

algorithm A is likely to return a bad hypothesis

No-Free-Lunch

- In other words, no Boolean concept learner can succeed on all learnable tasks — every learner will fail on some tasks where other learners succeed
- Key idea behind proof: every learner that sees less than half of all possible instances during training cannot be sure about the labels of the unseen instances, and may get all of them wrong
- We will not study the formal proof (*if interested, see the book or Shai Ben-Davis' video lecture*)

Prior Knowledge

- Successful learning needs to incorporate prior knowledge about the distribution D to avoid distributions causing failure, e.g.,
 - D comes from a specific parametric family of distributions (*we'll see examples in the second part of the module*)
 - Some h in a predefined class H has $L_D(h) = 0$ (*realisability*)
 - $\min_{h \in H} L_D(h)$ is small for predefined class H

Bias-Complexity Tradeoff

- For a given learning task, we'd like to choose H that allows for small error, but if we make H too large, learning fails.

approximation error: the error due to choosing this H (inductive bias)

- Let h_S be an ERM_H hypothesis, set $\epsilon_{app} = \min_{h \in H} L_D(h)$ and

$$\epsilon_{est} = L_D(h_S) - \epsilon_{app}$$

estimation error: difference between true error achieved by ERM and best possible error in H

Which classes H provide a good balance?

Which H are PAC-learnable*?

- All finite classes are PAC-learnable
- What about infinite classes?
- $H_{thr} = \{h_{\leq i} \mid i \in \mathbb{R}\}$ with $h_{\leq i}(x) = 1$ if $x \leq i$ and 0 otherwise **is PAC-learnable** using ERM
- $H_{fin} = \{h_M \mid (M \subseteq \mathbb{R} \wedge |M| < \infty) \vee M = \mathbb{R}\}$ with $h_M(x) = 1$ if $x \in M$ and 0 otherwise **is not PAC-learnable** using ERM


Key difference


- Intuitively, for every finite sample, H_{fin} contains a hypothesis that overfits to that sample, while this is not the case for H_{thr}
- This is formalised by the VC-dimension (named after Vapnik and Chervonenkis)

VC-Dimension

- Let H be a class of functions from X to $\{0,1\}$ and $C = \{c_1, \dots, c_m\}$ a finite subset of X . The **restriction** H_C of H to C contains exactly those functions from C to $\{0,1\}$ that agree with some h in H on C .
- A hypothesis class H **shatters** a finite subset C of X if the restriction of H to C contains all functions from C to $\{0,1\}$.
- The **VC-dimension** $\text{VCdim}(H)$ of a hypothesis class H is the maximal size of a set C that can be shattered by H . $\text{VCdim}(H)$ is infinite if H can shatter arbitrarily large sets.

VC-Dimension

- To show that the VC-dimension of class H is d , we need to show that
 - there is a set C of size d that is shattered by H 

providing an example and showing it is shattered is enough
 - every set C of size $d+1$ is **not** shattered by H 

requires proving that whatever C of size $d+1$ we choose, it is not shattered

Example

- $H_{thr} = \{h_{\leq i} \mid i \in \mathbb{R}\}$ with $h_{\leq i}(x) = 1$ if $x \leq i$ and 0 otherwise
- If C contains a single point c , there are two functions from C to $\{0,1\}$, and both agree with some h in H_{thr} (why?). Thus H_{thr} shatters C and $VCdim(H_{thr}) \geq 1$.
- If C contains two different points c_1 and c_2 , there are four functions from C to $\{0,1\}$, but only three of them agree with some h in H_{thr} (why?). Thus no set of size two is shattered by H_{thr} and $VCdim(H_{thr}) = 1$.

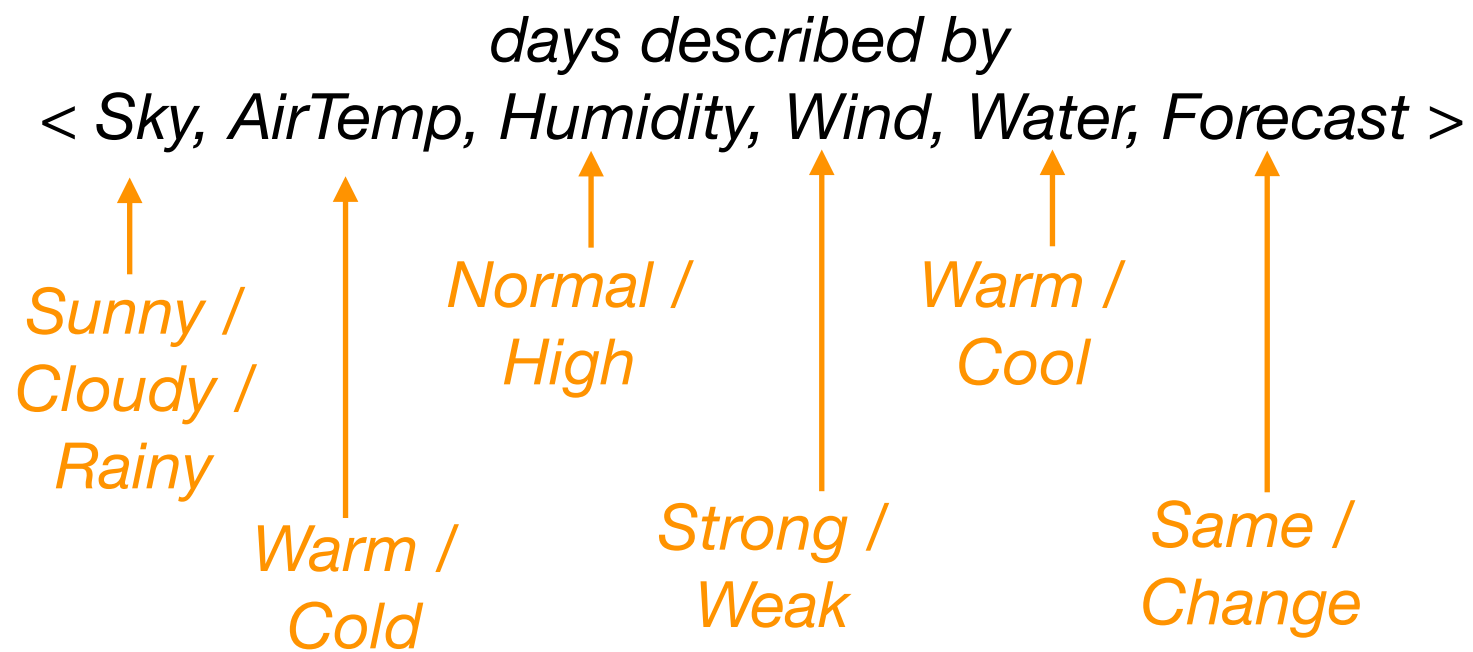
Example

- $H_{fin} = \{h_M \mid (M \subseteq \mathbb{R} \wedge |M| < \infty) \vee M = \mathbb{R}\}$ with $h_M(x) = 1$ if $x \in M$ and 0 otherwise
- Consider $C = \{c_1, \dots, c_m\}$ for some finite m . There are 2^m functions from C to $\{0,1\}$, and all agree with some H in H_{fin} (why?).
- Thus H_{fin} shatters arbitrarily large sets, and $VCdim(H_{fin}) = \infty$

VC-Dimension

- If H is finite, $VCdim(H) \leq \log_2(|H|)$ (*why?*)
- If H shatters some set C of size $2m$ then we cannot learn H using m examples.
- If H has infinite VC-dimension, then H is not PAC-learnable.

Example



- Let H be our earlier hypothesis space for this setting.
- $VCdim(H) \leq \log_2(|H|) = \log_2(973) = 9.93$
- $VCdim(H) \geq 6$ because H shatters the following set of six examples:
 - < cloudy, warm, normal, strong, warm, same >
 - < sunny, cold, normal, strong, warm, same >
 - < sunny, warm, high, strong, warm, same >
 - < sunny, warm, normal, weak, warm, same >
 - < sunny, warm, normal, strong, cool, same >
 - < sunny, warm, normal, strong, warm, change >

Fundamental Theorem of Statistical Learning

Let H be a class of functions from X to $\{0,1\}$. Then, the following are equivalent:

- H has finite VC-dimension.
- H is PAC-learnable.
- Any ERM learner is a successful PAC learner for H .
- H is agnostic PAC-learnable.
- Any ERM learner is a successful agnostic PAC learner for H .

Fundamental Theorem of Statistical Learning

- There is a quantitative version of this theorem for classes with finite VC-dimension that provides both lower and upper bounds on the sample complexity.
- For finite H , the lower bound on the sample complexity for PAC learning grows linearly in $\text{VCdim}(H)$ compared to logarithmically in the size of H .

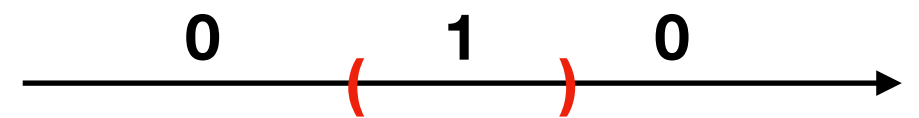
e.g., for (integer) threshold functions on $X=\{1,\dots,k\}$, $\text{VCDim}(H)=1$ but $|H|=k$

Exercise: determine the VC-dimension for each of these classes

- $H_{int} = \{h_{a,b} \mid a, b \in \mathbb{R}, a < b\}$ where

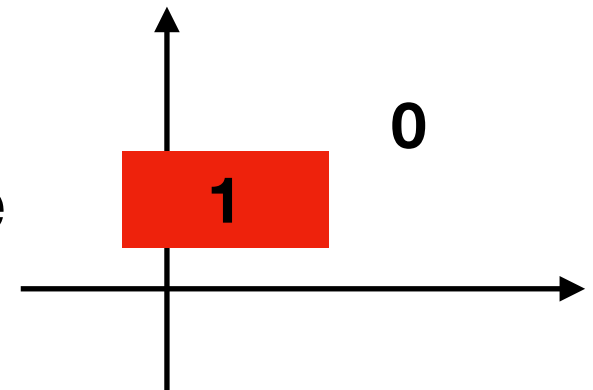
$$h_{a,b}(x) = \begin{cases} 1 & \text{if } x \in (a, b) \\ 0 & \text{otherwise} \end{cases}$$

open intervals on the real line



- $H_{rect} = \{h_{(a,b,c,d)} \mid a, b, c, d \in \mathbb{R}, a \leq b, c \leq d\}$ where

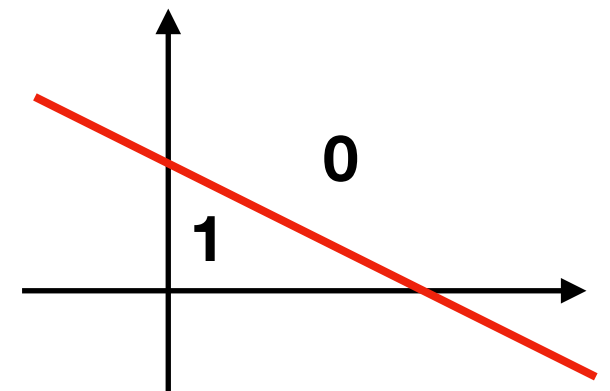
$$h_{(a,b,c,d)}(x, y) = \begin{cases} 1 & \text{if } a \leq x \leq b \text{ and } c \leq y \leq d \\ 0 & \text{otherwise} \end{cases}$$



axis-aligned rectangles on the real plane

- $H_{lin} = \{h_{(a,b,\theta)} \mid a, b \in \mathbb{R}, \theta \in \{ \leq, \geq \} \}$ where

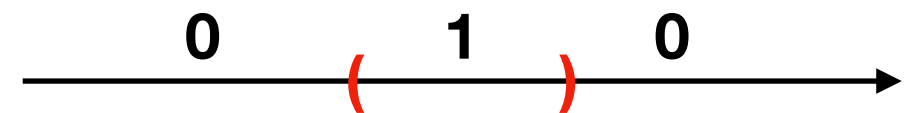
$$h_{(a,b,\theta)}(x, y) = \begin{cases} 1 & \text{if } (ax + b)\theta y \\ 0 & \text{otherwise} \end{cases}$$



separating lines on the real plane

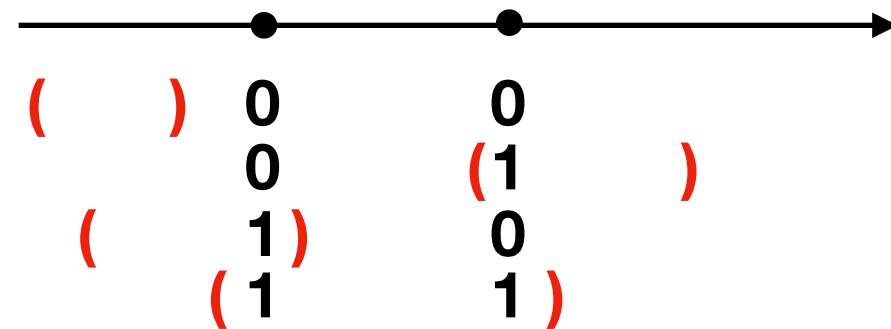
$$H_{int} = \{h_{a,b} \mid a, b \in \mathbb{R}, a < b\} \text{ where } h_{a,b}(x) = \begin{cases} 1 & \text{if } x \in (a, b) \\ 0 & \text{otherwise} \end{cases}$$

open intervals on the real line

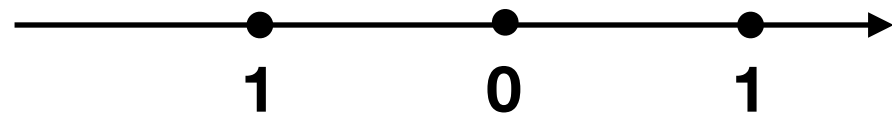


$$VCdim(H_{int}) = 2$$

example set of size 2 that is shattered



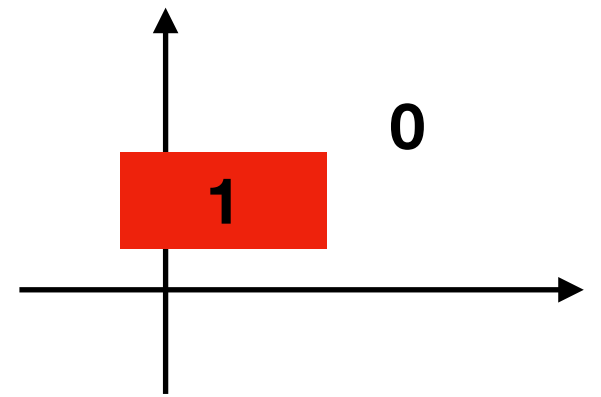
no set of size 3 can be shattered



no element of h can represent this

$$H_{rect} = \{h_{(a,b,c,d)} \mid a, b, c, d \in \mathbb{R}, a \leq b, c \leq d\} \text{ where}$$

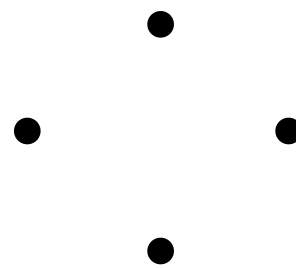
$$h_{(a,b,c,d)}(x, y) = \begin{cases} 1 & \text{if } a \leq x \leq b \text{ and } c \leq y \leq d \\ 0 & \text{otherwise} \end{cases}$$



axis-aligned rectangles on the real plane

$$VCdim(H_{rect}) = 4$$

example set of size 4 that is shattered



no set of size 5 can be shattered

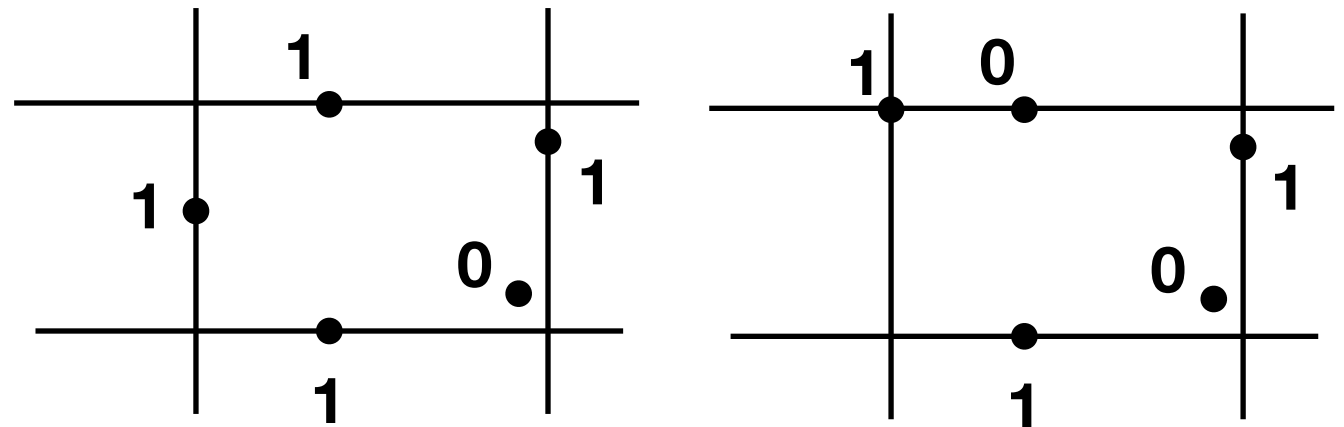
select a highest point and label it 1

select a lowest point and label it 1

select a leftmost point and label it 1

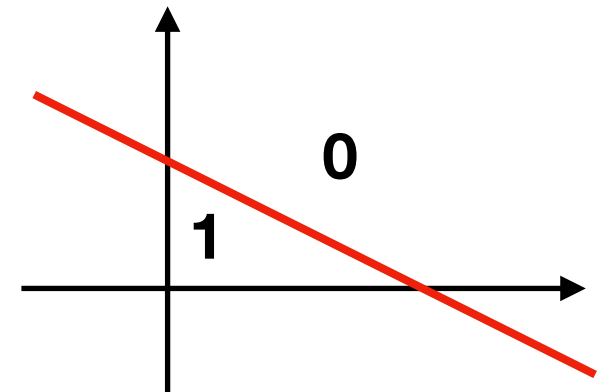
select a rightmost point and label it 1

label unlabeled points 0



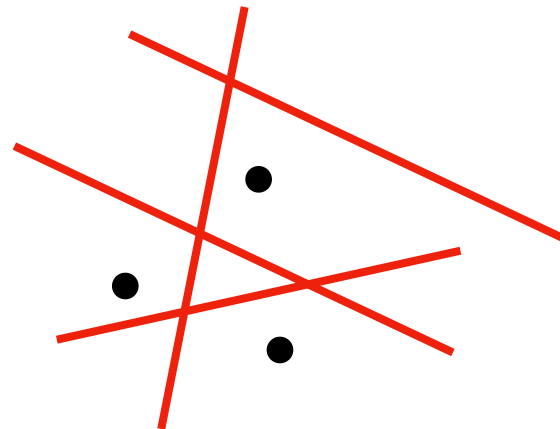
$$H_{lin} = \{h_{(a,b,\theta)} \mid a, b \in \mathbb{R}, \theta \in \{ \leq, \geq \} \} \text{ where } h_{(a,b,\theta)}(x, y) = \begin{cases} 1 & \text{if } (ax + b)\theta y \\ 0 & \text{otherwise} \end{cases}$$

separating lines on the real plane



$$VCdim(H_{lin}) = 3$$

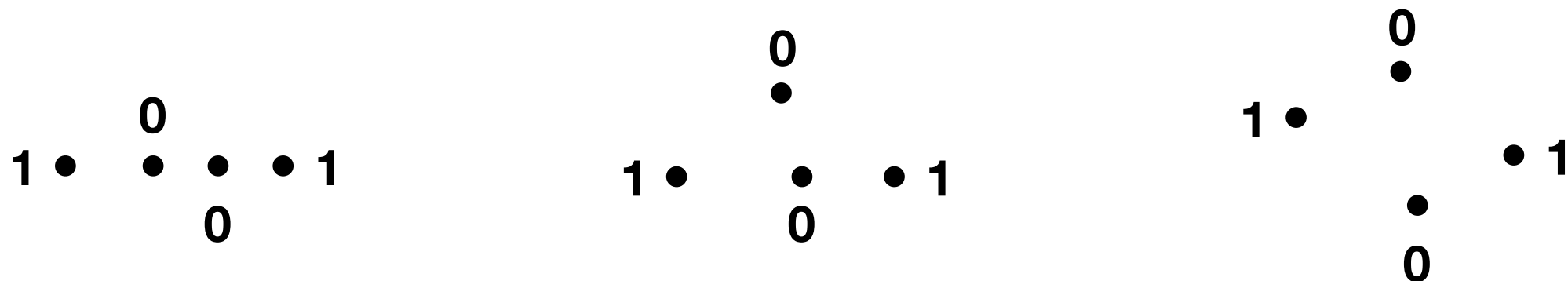
example set of size 3 that is shattered



no set of size 4 can be shattered

label a leftmost and a rightmost point 1 and the remaining two 0

(in case of vertically aligned points, label the highest & lowest points 1 and the other two 0)



Reading material

- Understanding machine learning: parts of
 - chapter 2 for ERM & finite H
 - chapter 3 for PAC & agnostic PAC
 - chapter 5 for no-free-lunch
 - chapter 6 for VC-dimension & fundamental theorem
- next time (in two weeks!) we'll discuss the **computational** complexity of learning (chapter 8)