Învățare Automată (Machine Learning)

Bogdan Alexe,

bogdan.alexe@fmi.unibuc.ro

Master Informatică, anul I, 2018-2019, cursul 3

Organizare

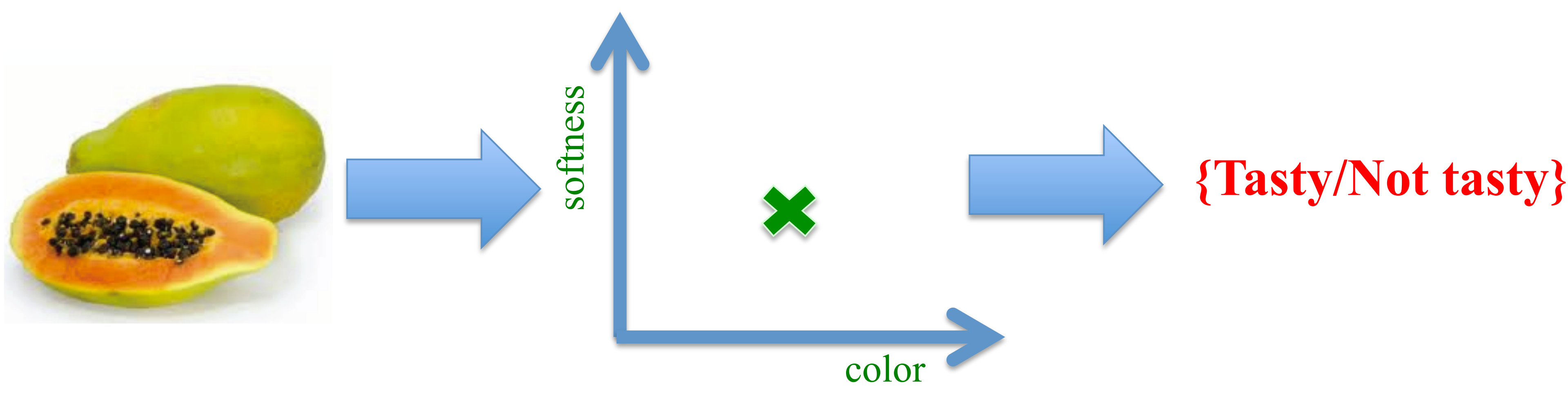
- am adăugat în orar o ședință de seminar, miercuri, săptămâna impară, între 16-18, sala 5

- azi la seminar facem ce am făcut săptămâna trecută și miercuri

- seminarul 2 - săptămâna viitoare

Recap

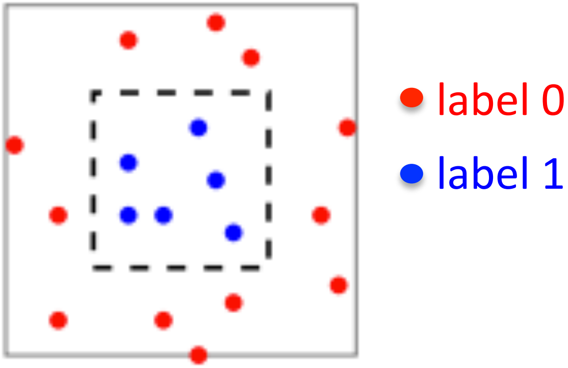
- A Formal Model – The Statistical learning framework
  - papaya tasting learning scenario, classification task: tasty – label 1, not tasty – label 0
  - domain set $\mathcal{X}$, label set $\mathcal{Y}$, training data S, prediction rule $h : \mathcal{X} \rightarrow \mathcal{Y}$
  - empirical error, generalization error
  - data generation model: i.i.d + realizability ("correct" labeling function, $f : \mathcal{X}$ –> $\mathcal{Y}$)



Recap

- Empirical Risk Minimization
  - learning paradigm that returns a predictor h that minimizes the empirical error on sample S
  - might overfit: small error on the training data, large error on the other samples

$$h_S(x) = \begin{cases} y_i & \text{if } \exists i \in [m] \text{ s.t. } x_i = x \\ 0 & \text{otherwise.} \end{cases}$$



  - $L_S(h_S) = 0$, but $L_{\mathcal{D},f}(h_S) = ½$ (h predicts the label 1 on a finite number of instances)
  - inductive bias: use prior knowledge and choose a hypothesis class $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$
  - apply the ERM learning paradigm over $\mathcal{H}$

Recap

- Probably Approximately Correct learning
  - can only be approximately correct: happy to find h_S with $L_{(\mathcal{D},f)}(h_S) \le \varepsilon$, where $\varepsilon \in (0, 1)$ is the accuracy parameter, user-specified
  - can only be probably correct: allow the algorithm to fail with probability δ, where $\delta \in (0, 1)$ is the confidence parameter, user-specified
  - definition of PAC learnability of hypothesis class $\mathcal{H}$ in the realizability case

PAC learnability of a class $\mathcal{H}$

A hypothesis class $\mathcal{H}$ is called ***PAC learnable*** if there exists a function $m_{\mathcal{H}}: (0,1)^2 \rightarrow N$ and a learning algorithm A with the following property:

- for every $\varepsilon > 0$ (*accuracy* → *"approximately correct"*)
- for every $\delta > 0$ (*confidence* → *"probably"*)
- for every labeling $f \in \mathcal{H}$ (*realizability case*)
- for every distribution $\mathcal{D}$ over $\mathcal{X}$

when we run the learning algorithm A on a training set S, consisting of $m \geq m_{\mathcal{H}}(\varepsilon, \delta)$ examples sampled i.i.d. from $\mathcal{D}$ and labeled by f the algorithm A returns a hypothesis $h_S \in \mathcal{H}$ such that, with probability at least $1-\delta$ (over the choice of examples), $L_{D,f}(h_S) \leq \varepsilon$.

$$\underset{S \sim D^m}{P}(L_{f,D}(h_S) \leq \varepsilon) \geq 1-\delta$$

- $h_S = A(S)$
- the function $m_{\mathcal{H}}: (0,1)^2 \rightarrow N$ is called sample complexity of learning $\mathcal{H}$
- $m_{\mathcal{H}}(\varepsilon, \delta)$ – the minimum number of examples required to guarantee a PAC solution

L. G. Valiant, *A theory of the Learnable*, Communications ACM, 27(11):1134-1142, 1984

PAC learnability of a class $\mathcal{H}$

A hypothesis class $\mathcal{H}$ is called ***PAC learnable*** if there exists a function $m_{\mathcal{H}}: (0,1)^2 \rightarrow N$ and a learning algorithm A with the following property:

- for every $\varepsilon > 0$ (*accuracy* → *"approximately correct"*)
- for every $\delta > 0$ (*confidence* → *"probably"*)
- for every labeling $f \in \mathcal{H}$ (*realizability case*)
- for every distribution $\mathcal{D}$ over $\mathcal{X}$

when we run the learning algorithm A on a training set S, consisting of $m \geq m_{\mathcal{H}}(\varepsilon, \delta)$ examples sampled i.i.d. from $\mathcal{D}$ and labeled by f the algorithm A returns a hypothesis $h_S \in \mathcal{H}$ such that, with probability at least $1-\delta$ (over the choice of examples), $L_{\mathrm{D},f}(h_S) \leq \varepsilon$.

$$\underset{S\sim D^m}{P}(L_{f,D}(h_S) \leq \varepsilon) \geq 1-\delta \Leftrightarrow \underset{S\sim D^m}{P}(L_{f,D}(h_S) > \varepsilon) < \delta$$

L. G. Valiant, *A theory of the Learnable*, Communications ACM, 27(11):1134-1142, 1984

Learning finite classes

Theorem:

Finite hypothesis classes $\mathcal{H}$ are PAC-learnable.

Idea of the proof

- a bad predictor h_b has $L_{D,f}(h_b) > \varepsilon$
- h_b can be output by the $ERM_{\mathcal{H}}$ learning paradigm if has zero empirical error: $L_S(h_b) = 0$
- this can happen if h_b labels correctly all the m training examples from S i.i.d from $\mathcal{D}$
- given a random example from $\mathcal{D}$, h_b has $< 1\text{-}\varepsilon$ probability to label it correctly
- h_b labels correctly all the m training examples from S with probability $< (1\text{-}\varepsilon)^m \leq e^{-\varepsilon m}$
- there are at most $|\mathcal{H}|$ bad hypthotesis, so consider $|H| \times e^{-\varepsilon m} \leq \delta$, so take $m \geq \dfrac{\log(|\mathcal{H}|/\delta)}{\epsilon}$

Concept class

- h: $\mathcal{X} \rightarrow \{0,1\}$ the *target concept* to learn
  - can be identified with its support $\{x \in \mathcal{X} \mid h(x) = 1\}$
  - set of points inside a rectangle
    - h = indicator function of these points
    - the concept to learn is a rectangle

- $\mathcal{H}$ can be interpreted as the concept class, a set of target concepts h
  - set of all rectangles in the plane
  - conjunction of Boolean literals

Conjunctions of Boolean literals

- C_n = concept class of conjunctions of at most n Boolean literals $x_1, \ldots, x_n$
  - a Boolean literal is either x_i or its negation $\overline{x_i}$
  - can interpret x_i as feature i
  - example: $h = x_1 \wedge \overline{x_2} \wedge x_4$ where $\overline{x_2}$ denotes the negation of the Boolean literal x_2

- observe that for n = 4:
  - a positive example such as (1, 0, 0, 1) implies that the target concept cannot contain the literals $\overline{x_1}$, x_2, x_3 and $\overline{x_4}$
    - for example if x_2 was present in the conjunction then for the current positive example (where x_2 has value 0) the label should have been 0
  - cannot say anything about literals x_1, $\overline{x_2}$, $\overline{x_3}$ and x_4. They might be present or absent in the conjunction (target concept) that we are searching for
  - the first positive example eliminates half of the literals

  - in contrast, a negative example such as (1, 0, 0, 0) is not as informative since it is not known which of its n bits are incorrect.

Conjunctions of Boolean literals

- C_n = concept class of conjunctions of at most n Boolean literals $x_1, \ldots, x_n$
- a simple algorithm for finding a consistent hypothesis is thus based on positive examples and consists of the following:
  - for each positive example $(b_1, \ldots b_n)$,
    - if $b_i = 1$ then $\overline{x_i}$ is ruled out as a possible literal in the concept class
    - if $b_i = 0$ then x_i is ruled out.
  - the conjunction of all the literals not ruled out is thus a hypothesis consistent with the target

0	1	1	0	1	1	+
0	1	1	1	1	1	+
0	0	1	1	0	1	-
0	1	1	1	1	1	+
1	0	0	1	1	0	-
0	1	0	0	1	1	+

$x_1 \quad \overline{x_1} \quad x_2 \quad \overline{x_2} \quad x_3 \quad \overline{x_3} \quad x_4 \quad \overline{x_4} \quad x_5 \quad \overline{x_5} \quad x_6 \quad \overline{x_6}$

$$\rightarrow \overline{x}_1 \wedge x_2 \wedge x_5 \wedge x_6$$

Conjunctions of Boolean literals

- C_n = concept class of conjunctions of at most n Boolean literals $x_1, \ldots, x_n$
- $|C_n| = 3^n$ – finite, so is PAC learnable with sample complexity $m_{\mathcal{H}}(\varepsilon, \delta) \leq m$:

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

$$m \geq \left\lceil \frac{1}{\varepsilon}\left(n\log(3) + \log(\frac{1}{\delta})\right)\right\rceil$$

$$m \geq \left\lceil \frac{1}{\varepsilon}\left(n\log(3) - \log(\delta)\right)\right\rceil$$

- for $\varepsilon = 0.01$, $\delta = 0.02$, $n = 10$, $m \geq 149$, no matter how $\mathcal{D}$ looks like, all possible examples are $2^{10} = 1024$
- we need at least 149 examples; the bound guarantees (at least) 99% accuracy with (at least) 98% confidence

Universal concept class $\mathcal{U}_n$

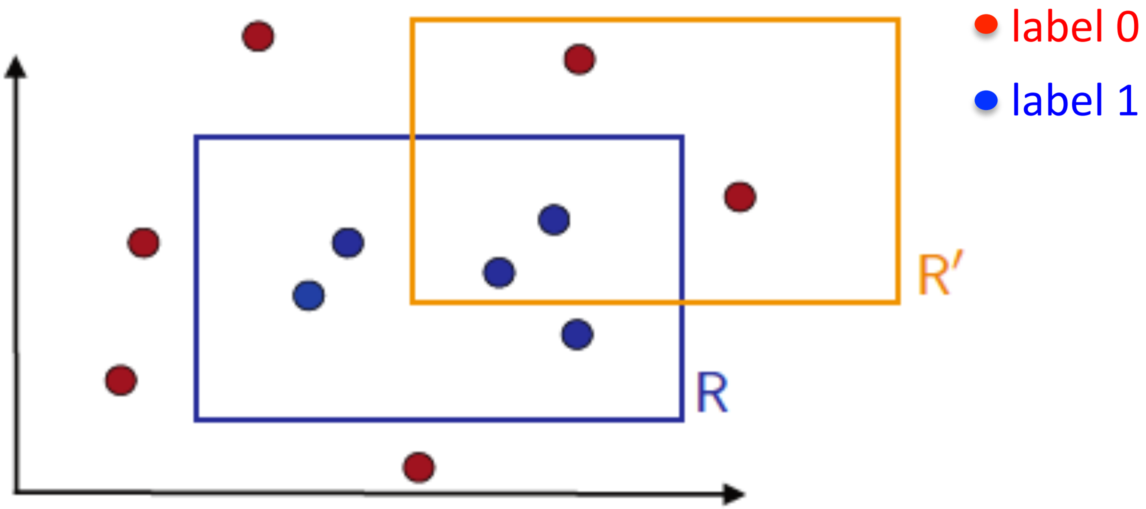
- B^n = set of boolean n-tuples, $|B| = 2^n$
- want to learn arbitrary subsets of B^n
- $\mathcal{U}_n = \{h\colon B^n \rightarrow \{0,1\}\}$ - the concept class formed by all subsets of B^n
- $\mathcal{U}_n$ – universal class
- is this concept class PAC-learnable?
- $|\mathcal{U}_n| = 2^{2^n}$ – finite, so is PAC learnable with $m_{\mathcal{H}}(\varepsilon, \delta)$ in the order of m:

$$m \geq \left\lceil \frac{1}{\varepsilon}\left(2^n \log(2) + \log(\frac{1}{\delta})\right)\right\rceil$$

- sample complexity exponential in n, number of variables
- $\mathcal{U}_n$ is finite and hence PAC-learnable, but we will need exponential time (to inspect exponentially many examples)
- for $\varepsilon = 0.01$, $\delta = 0.02$, n = 10, $m \geq 71370$, no matter how $\mathcal{D}$ looks like, all possible examples are $2^{10} = 1024$
- it is not PAC-learnable in any practical sense (need polynomial time complexity = later require $m_{\mathcal{H}}$ be polynomial in $1/\varepsilon, 1/\delta, n, |\mathcal{H}|$)

Axis-aligned rectangles

- $\mathcal{X} = \mathrm{R}^2$ points in the plane
- $\mathcal{H}$ = set of all axis-aligned rectangle lying in R^2
- each concept $h \in \mathcal{H}$ is an indicator function of a rectangle
- the learning problem consists of determining with small error a target axis-aligned rectangle using the labeled training sample



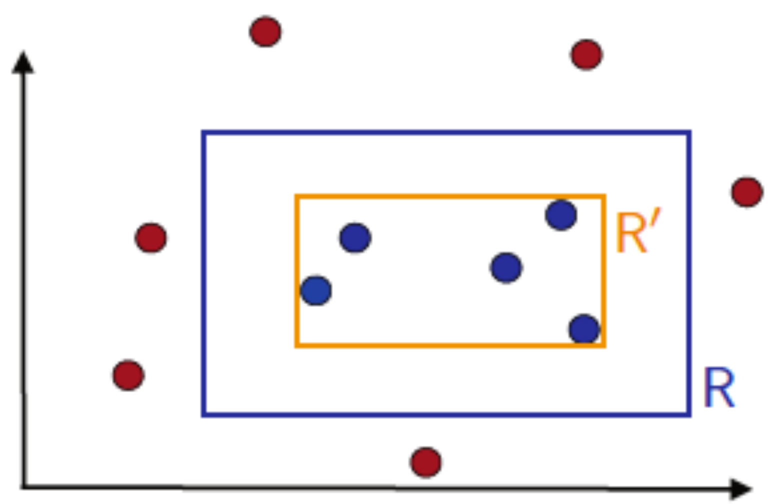
Target concept R and possible hypothesis R′. Circles represent training instances. A blue circle is a point labeled with 1, since it falls within the rectangle R. Others are red and labeled with 0.

Axis-aligned rectangles

- $\mathcal{X} = \mathrm{R}^2$ points in the plane
- $\mathcal{H}$ = set of all axis-aligned rectangle lying in R^2
- $|\mathcal{H}| = \infty$
- still $\mathcal{H}$ is PAC-learnable with sample complexity in the order of:

$$m \geq \left\lceil \frac{4}{\varepsilon} \log\left(\frac{1}{\delta}\right) \right\rceil$$

- simple algorithm: take the tightest rectangle enclosing all the positive examples (or take the largest rectangle not including negative samples)
- discuss this example in seminar



Today's lecture: Overview

- The general Probably Approximately Correct learning model
- Uniform Convergence for Agnostic PAC learnability

The general PAC model

Relaxing the realizability assumption – Agnostic PAC learning

- so far we assumed that labels are generating by some $f \in \mathcal{H}$
  - f is a function: the features fully determines the label (two papayas with the same color and softness will have the same label)
  - f is in $\mathcal{H}$, e.g. there is a rectangle in the color-softness space that determines the labels of papayas

- this assumption may be too strong

- relax the realizability assumption by replacing the "target labeling function" with a more flexible notion, a data-labels generating distribution.
  - f might not be a function
  - f might not be in $\mathcal{H}$ (inconsistency case)

Relaxing the realizability assumption – Agnostic PAC learning

- recall: in the PAC model, $\mathcal{D}$ is a distribution over $\mathcal{X}$
  - if example x appears in the training data it has a fixed label

- consider from now on that $\mathcal{D}$ is a distribution over $\mathcal{X} \times \mathcal{Y}$
  - if example x appears in the training data it might have a different label each time

- redefine the risk = generalization error as:

$$L_{\mathcal{D}}(h) \stackrel{\text{def}}{=} \underset{(x,y)\sim\mathcal{D}}{\mathbb{P}}[h(x) \neq y] \stackrel{\text{def}}{=} \mathcal{D}(\{(x,y) : h(x) \neq y\})$$

- redefine the "approximately correct" notion to:

$$L_{\mathcal{D}}(A(S)) \leq \min_{h\in\mathcal{H}} L_{\mathcal{D}}(h) + \epsilon$$

$A(S) = h_S$ is ε-accurate wrt $\mathcal{D}$, $\mathcal{H}$

PAC vs. Agnostic PAC learning

	PAC	Agnostic PAC

The Bayes optimal predictor

- given any probability distribution $\mathcal{D}$ over $\mathcal{X} \times \{0,1\}$, the best label prediction function we can achieve is the Bayes rule:

$$f_{\mathcal{D}}(x) = \begin{cases} 1 & \text{if } \mathbb{P}[y=1|x] \geq 1/2 \\ 0 & \text{otherwise} \end{cases} \Leftrightarrow \mathcal{D}((x,1)|x) \geq ½$$

- for any probability distribution $\mathcal{D}$, the Bayes predictor $f_{\mathcal{D}}$ is optimal, in the sense that no other classifier g: $\mathcal{X} \rightarrow \{0,1\}$has a lower error, $L_{\mathcal{D}}(f_{\mathcal{D}}) \leq L_{\mathcal{D}}(g)$ (seminar exercise)

- we don't know the probability distribution $\mathcal{D}$ that produces the data (x, y), we only see a sample S generated by $\mathcal{D}$

- so, we cannot utilize the Bayes optimal predictor $f_{\mathcal{D}}$

1-NN predictor

- if we fix the data-generating distribution $\mathcal{D}$ and then let m go to infinity, then the error of the 1-NN rule converges to twice the Bayes error

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}(h_S)] \leq 2\,L_{\mathcal{D}}(h^\star) + c \mathop{\mathbb{E}}_{S\sim\mathcal{D}^m,\mathbf{x}\sim\mathcal{D}}[\|\mathbf{x}-\mathbf{x}_{\pi_1(\mathbf{x})}\|].$$

- the analysis can be generalized to larger values of k nearest neighbors, showing that the expected error of the k-NN rule converges to $1+\sqrt{8/k}$ times the error of the Bayes classifier

$$\mathop{\mathbb{E}}_{S}[L_{\mathcal{D}}(h_S)] \leq \left(1+\sqrt{\frac{8}{k}}\right) L_{\mathcal{D}}(h^\star) + \left(6c\sqrt{d}+k\right) m^{-1/(d+1)}.$$

Beyond binary classification

Scope of learning problems

- multiclass classification: $\mathcal{Y}$ is finite representing $|\mathcal{Y}|$ different classes. E.g. $\mathcal{X}$ is documents and $\mathcal{Y} = \{\text{News, Sports, Biology, Medicine}\}$

- regression: Y = R. E.g. one wishes to predict the stock price tomorrow, the max temperature, a baby's birth weight based on ultrasound measure of his head circumference, abdominal circumference and femur length
  - what is fundamental difference to multiclass classification?
  - the loss suffered when making a bad prediction

Loss functions

- let $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$

- given hypothesis $h \in \mathcal{H}$ and an example z = $(x,y) \in \mathcal{Z}$, how good is h on (x,y)?
- loss function $\ell : \mathcal{H} \times \mathcal{Z} \rightarrow \mathrm{R}_+$
  - measures the error that model h does it on the instance $z = (x,y)$
  - the true risk (generalization error) of model h is: $L_{\mathcal{D}}(h) \stackrel{\text{def}}{=} \mathop{\mathbb{E}}_{z\sim\mathcal{D}}[\ell(h,z)]$

- example: 0-1 loss: $\ell(h,(x,y)) = \begin{cases} 1 & \text{if } h(x) \neq y \\ 0 & \text{if } h(x) = y \end{cases}$ binary class prediction, multiclass prediction

$$\mathrm{E}_{z\sim\mathcal{D}}[\ell(h,z)] = \mathrm{E}_{(x,y)\sim\mathcal{D}}[\ell(h,(x,y)] = \mathrm{E}_{(x,y)\sim\mathcal{D}}[0 \times 1_{[h(x)=y]} + 1 \times 1_{[h(x)\neq y]}] =$$
$$= \mathrm{E}_{(x,y)\sim\mathcal{D}}[1_{[h(x)\neq y]}] = \mathcal{D}(\{(x,y) |\ h(x) \neq y\}) = \mathrm{P}_{(x,y)\sim\mathcal{D}}(h(x) \neq y)$$

Loss functions

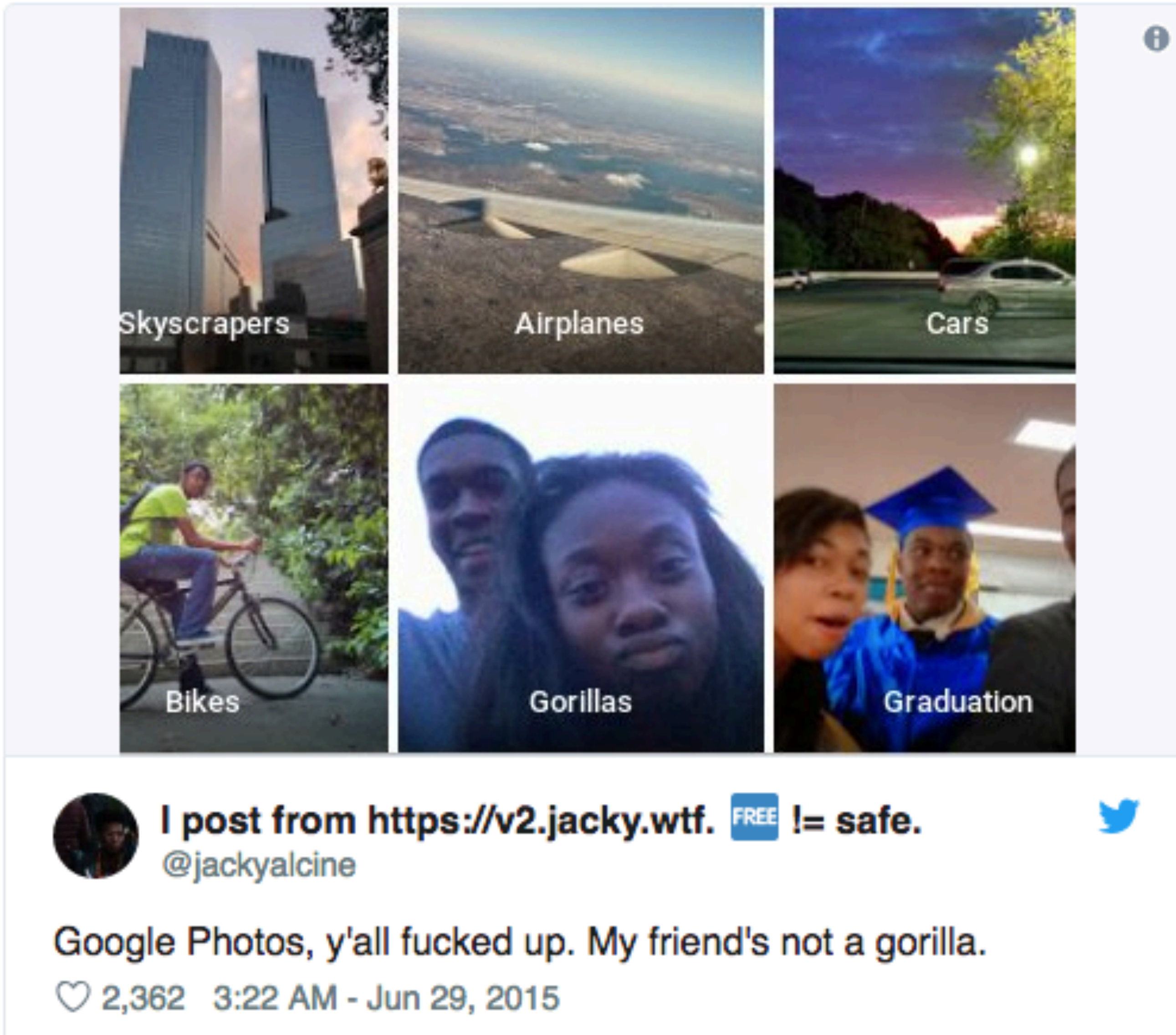
- let $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$

- given hypothesis $h \in \mathcal{H}$ and an example z = $(x,y) \in \mathcal{Z}$, how good is h on (x,y)?
- loss function $\ell : \mathcal{H} \times \mathcal{Z} \rightarrow \mathrm{R}_+$
  - measures the error that model h does it on the instance $z = (x,y)$
  - the true risk (generalization error) of model h is: $L_{\mathcal{D}}(h) \stackrel{\text{def}}{=} \mathop{\mathbb{E}}_{z \sim \mathcal{D}}[\ell(h, z)]$

- example of other loss functions:

Squared loss: $\ell(h,(x,y)) = (h(x) - y)^2$
Absolute-value loss: $\ell(h,(x,y)) = |h(x) - y|$
Cost-sensitive loss: $\ell(h,(x,y)) = C_{h(x),y}$ where C is some $|\mathcal{Y}| \times |\mathcal{Y}|$ matrix

Cost-sensitive loss



Cost-sensitive loss

GOOGLE \ TECH \ ARTIFICIAL INTELLIGENCE

Google 'fixed' its racist algorithm by removing gorillas from its image-labeling tech

Nearly three years after the company was called out, it hasn't gone beyond a quick workaround

By James Vincent | Jan 12, 2018, 10:35am EST

A spokesperson for Google confirmed to *Wired* that the image categories "gorilla," "chimp," "chimpanzee," and "monkey" remained blocked on Google Photos after Alciné's tweet in 2015. "Image labeling technology is still early and unfortunately it's nowhere near perfect," said the rep. The categories are still available on other Google services, though, including the Cloud Vision API it sells to other companies and Google Assistant.

The general PAC learning problem

- we wish to Probably Approximately solve:

$$\min_{h\in\mathcal{H}} L_{\mathcal{D}}(h) \quad \text{where} \quad L_{\mathcal{D}}(h) \stackrel{\text{def}}{=} \mathop{\mathbb{E}}_{z\sim\mathcal{D}}[\ell(h,z)]$$

- learner knows $\mathcal{H}$, $\mathcal{Z}$ and loss function ℓ
- learner receives accuracy parameter ε and confidence parameter δ
- learner can decide on training set size m based on ε, δ
- learner doesn't know $\mathcal{D}$ but can sample S from $\mathcal{D}^m$
- using S the learner outputs some hypothesis $A(S) = h_S$
- we want that with probability at least $1 - \delta$ over the choice of S, the following would hold:

$$L_{\mathcal{D}}(A(S)) \leq \min_{h\in\mathcal{H}} L_{\mathcal{D}}(h) + \epsilon$$

Formal definition

A hypothesis class $\mathcal{H}$ is called ***agnostic PAC learnable*** if there exists a function $m_{\mathcal{H}}: (0,1)^2 \rightarrow \mathbb{N}$ and a learning algorithm A with the following property:

- for every $\varepsilon > 0$ (*accuracy* → *"approximately correct"*)
- for every $\delta > 0$ (*confidence* → *"probably"*)
- for every distribution $\mathcal{D}$ over $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$

when we run the learning algorithm A on a training set S, consisting of $m \geq m_{\mathcal{H}}(\varepsilon, \delta)$ examples sampled i.i.d. from $\mathcal{D}$ the algorithm A returns a hypothesis A(S) from $\mathcal{H}$ such that, with probability at least $1-\delta$ (over the choice of examples) it holds that:

$$L_{\mathcal{D}}(A(S)) \leq \min_{h \in \mathcal{H}} L_{\mathcal{D}}(h) + \epsilon$$

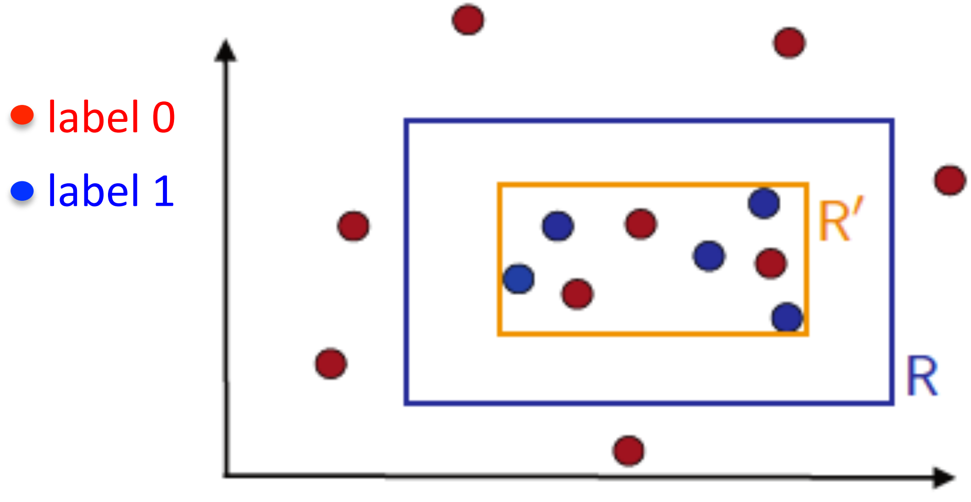
- if the realizability assumption holds, agnostic PAC = PAC
- in agnostic PAC learning, a learner can still declare success if its error is not much larger than the best error achievable by a predictor from the class $\mathcal{H}$.

Learning in the presence of noise - rectangles

- $\mathcal{X} = \mathrm{R}^2$ points in the plane
- $\mathcal{H}$ = set of all axis-aligned rectangle lying in R^2
- each concept $h \in \mathcal{H}$ is an indicator function of a rectangle
- the learning problem consists of determining with small error a target axis-aligned rectangle using the labeled training sample
- the training points received by the learner are subject to noise:
  - points negatively labeled are unaffected by noise
  - the label of a positive training points is randomly flipped to negative with probability $0 < \eta < ½$ (η is unknown)

$\mathcal{H}$ is agnostic PAC learnable

$\min_h L_{\mathcal{D}}(h) = \eta \times \mathcal{D}(\mathrm{R})$



A note of Caution

The fact that $\mathcal{H}$ is agnostically PAC learnable using the ERM paradigm doesn't mean that the result is any good.

It only means that you can be reasonable sure the ERM paradigm gives you a result that is close to the optimal result.

If the optimal result is bad (because, for example, the hypothesis class $\mathcal{H}$ fits the data really badly) the ERM paradigm will also give you a bad result.

PAC doesn't tell you that your hypothesis class $\mathcal{H}$ fits the data well, it only tells you that, if it fits well, the ERM paradigm will probably give you a reasonable good hypothesis.

Beyond the general PAC learning definition

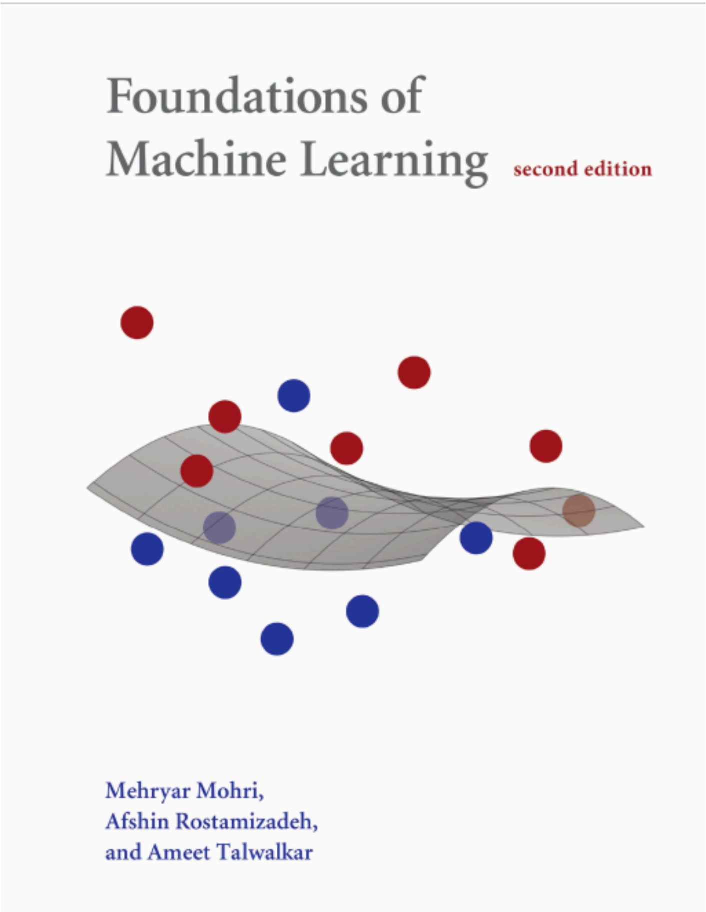
- the definition of the general PAC learning tells us:
  - when we consider we can learn something

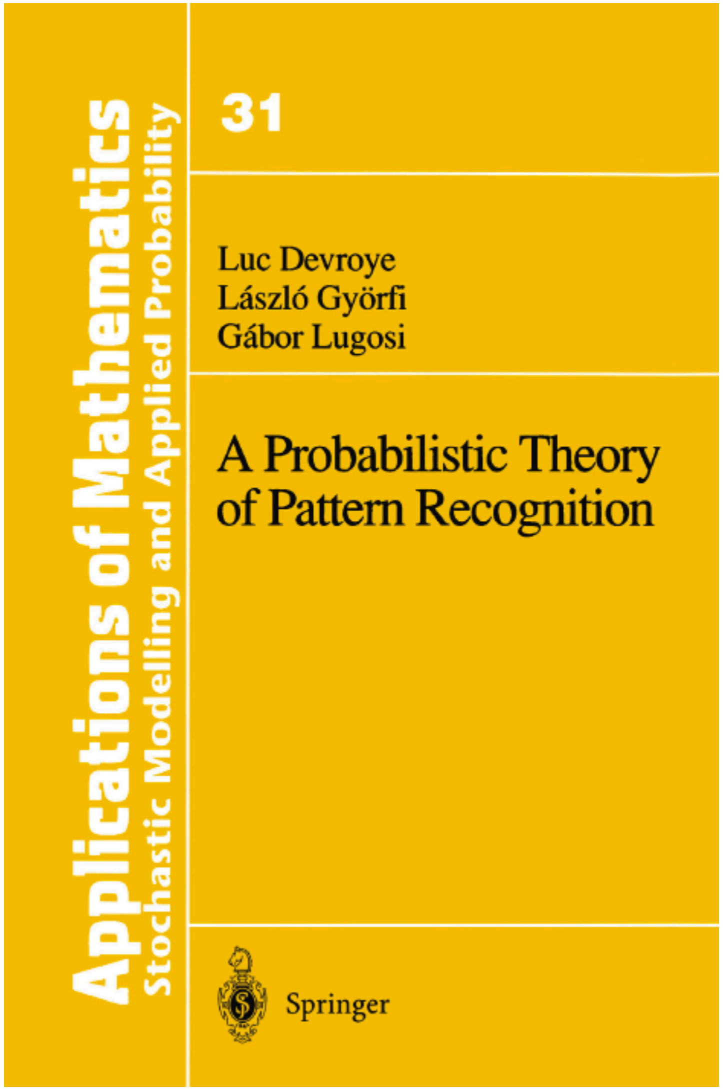
- the definition of the general PAC learning doesn’t tell us:
  - what we can learn
  - how we learn

- discover what can be general PAC-learned and how

Uniform Convergence – next time

Bibliography





Next time