Learning Reliable Rules under Class Imbalance

(Appeared in SDM21)

Dimitris Diochnos and Theodore Trafalis

University of Oklahoma

NSF AI Institute for Research on Trustworthy AI in Weather, Climate, and Coastal Oceanography (**AI2ES**)

September 29, 2021

Outline

Outline

Motivation

- Binary classification problems.
- Imbalanced datasets (rare events).
- The traditional learning framework has a 'naive' requirement for success: make few mistakes on average (low risk).
- In situations with extreme class imbalance we can just predict the majority class and we will have very low risk (error rate);
  e.g., predict that an extreme weather event (e.g., a tornado) is not going to happen in any given location.
- But this is not what we really want!

Motivation

- Binary classification problems.
- Imbalanced datasets (rare events).
- The traditional learning framework has a 'naive' requirement for success: make few mistakes on average (low risk).
- In situations with extreme class imbalance we can just predict the majority class and we will have very low risk (error rate);
  e.g., predict that an extreme weather event (e.g., a tornado) is not going to happen in any given location.
- But this is not what we really want!

How can we measure the performance of learning systems when we want to predict rare events?

Motivation

- Binary classification problems.
- Imbalanced datasets (rare events).
- The traditional learning framework has a 'naive' requirement for success: make few mistakes on average (low risk).
- In situations with extreme class imbalance we can just predict the majority class and we will have very low risk (error rate); e.g., predict that an extreme weather event (e.g., a tornado) is not going to happen in any given location.
- But this is not what we really want!

How can we measure the performance of learning systems when we want to predict rare events?

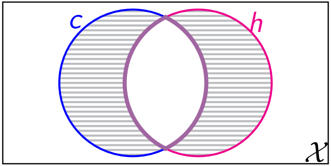
- We use primarily **two metrics beyond low risk**:
  - Recall
  - Precision

Representative Related Work

- Under-sampling the majority class; e.g., [Liu, Wu, and Zhou, 2009]
- Creation of synthetic data and over-sampling the minority class (**SMOTE**); [Chawla et al., 2002]
- Sampling based on clusters; [Jo and Japkowicz, 2004]
- Custom modification of established methods; e.g., SVMs [Wu and Chang, 2004] or boosting [Sun et al., 2007]
- Reweighting; [Wang, Ramanan, and Hebert, 2017]
- Margin-based methods; [Cao et al., 2019]
- Complex performance measures; [Joachims, 2005; Narasimhan et al., 2015]

Probably Approximately Correct (PAC) Learning

- There is an *arbitrary, unknown* distribution D over $\mathcal{X}$.
- Learn from *poly* $\left(\frac{1}{\varepsilon}, \frac{1}{\delta}\right)$ many examples $(x, c(x))$, where $x \sim D$.
- The risk is defined as $\boldsymbol{R_D}(\boldsymbol{h}, \boldsymbol{c}) = \Pr_{x \sim D}(h(x) \neq c(x))$.



Goal 1 (Valiant, 1984)

$$\Pr_{S \sim D^m}\left(R_D(h, c) \leq \varepsilon\right) \geq 1 - \delta\ .$$

Definition 1 (Realizable Learning Problem)

A learning problem $(\mathcal{X}, \mathcal{C}, \mathcal{H}, \mathcal{D})$ is said to be realizable, if for any $D \in \mathcal{D}$ and any $c \in \mathcal{C}$, there exists at least one $h \in \mathcal{H}$ such that $R_D(h, c) = 0$.

Recall and Precision

Definition 2 (Recall and Precision)

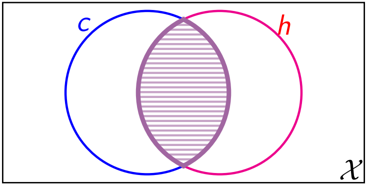
Given a hypothesis $h \in \mathcal{H}$, a target concept $c \in \mathcal{C}$, and an underlying distribution D, we have:

- the *recall* of h is defined by

$$\mathrm{Rec}_D\left(h, c\right) = \mathrm{Pr}_{x \sim D}\left(h(x) = 1 \mid c(x) = 1\right) \ .$$

- the *precision* of h is defined by

$$\mathrm{Prec}_D\left(h, c\right) = \mathrm{Pr}_{x \sim D}\left(c(x) = 1 \mid h(x) = 1\right) \ .$$



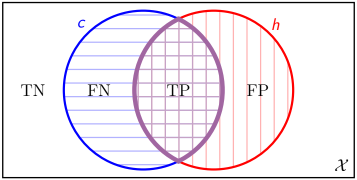
Empirical Recall and Precision

Definition 3 (Empirical Recall)

$$\widehat{\mathrm{Rec}}_S(h, c) = \frac{TP}{TP + FN}\,.$$

Definition 4 (Empirical Precision)

$$\widehat{\mathrm{Prec}}_S(h, c) = \frac{TP}{TP + FP}\,.$$



PAC Learning in the Realizable Case

Theorem 5 (Blumer et al, 1987)

Let $\mathcal{H}$ be a finite hypothesis class. Under the realizability assumption, a concept class $\mathcal{C}$ is PAC-learnable by $\mathcal{H}$ with sample complexity $m \leq \left\lceil \frac{1}{\varepsilon} \cdot \ln\left(\frac{|\mathcal{H}|}{\delta}\right)\right\rceil$.

Theorem 6

Let $\mathcal{H}$ be a hypothesis class with $\text{VC-dim}(\mathcal{H}) = d < \infty$. Under the realizability assumption, a concept class $\mathcal{C}$ is PAC-learnable by $\mathcal{H}$ with sample complexity

- $m \leq \mathcal{O}\left(\frac{1}{\varepsilon} \cdot (d \ln(1/\varepsilon) + \ln(1/\delta))\right)$ *[Vapnik & Chervonenkis, 1974; Blumer et al., 1989]*
- $m \leq \mathcal{O}\left(\frac{1}{\varepsilon} \cdot (d + \ln(1/\delta))\right)$ *[Hanneke, 2016]*

Outline

Summary of our Contributions

1. We extend the Probably Approximately Correct (PAC) model of learning and also include explicitly high recall and high precision among its goals at the end of the learning process.
2. We give lower bounds on the recall and the precision of a learned hypothesis based on its risk and the rate of the minority class.
3. An algorithm to obtain a lower bound on the rate of the minority class.
4. $\mathcal{C}$ is PAC learnable $\Rightarrow$ $\mathcal{C}$ is PAC learnable with high recall and high precision.
5. Experimental evaluation by studying two algorithms for learning monotone conjunctions under the uniform distribution.
   (source code: https://github.com/diochnos/pac-imbalanced)

PAC Learning Extension

Goal 1 (Valiant, 1984)

$$\Pr_{S \sim D^m}\left(R_D\left(h, c\right) \leq \varepsilon\right) \geq 1 - \delta \ .$$

Goal 2 (Our Extension of the PAC Learning Framework)

$$\Pr_{S \sim D^m}\left(\begin{array}{lcl} \left(R_D\left(h, c\right)\right. & \leq & \left.\varepsilon\right) \\ \wedge \left(\mathrm{Rec}_D\left(h, c\right)\right. & \geq & \left.1 - \gamma\right) \\ \wedge \left(\mathrm{Prec}_D\left(h, c\right)\right. & \geq & \left.1 - \xi\right) \end{array}\right) \geq 1 - \delta \, .$$

Lower Bounds on the Recall and the Precision

Proposition 1 (Lower Bound for Recall)

Let p_b be given such that $\Pr_{x\sim D}(c(x)=1) \geq p_b > 0$. Let $h \in \mathcal{H}$ be a hypothesis with risk $R_D(h,c)$. Then, for this hypothesis h it holds

$$\mathrm{Rec}_D(h,c) \geq 1 - \frac{R_D(h,c)}{p_b}\,.$$

Proposition 2 (Lower Bound for Precision)

Let p_b be given such that $\Pr_{x\sim D}(c(x)=1) \geq p_b > 0$. Let $h \in \mathcal{H}$ be a hypothesis with risk $R_D(h,c)$ and for which it holds $\mathrm{Rec}_D(h,c) \geq 1-\gamma$ for some $0 \leq \gamma < 1$. Then, for this hypothesis h it holds

$$\mathrm{Prec}_D(h,c) \geq 1 - \frac{R_D(h,c)}{(1-\gamma)p_b}\,.$$

Implications

Theorem 7 (Informal)

Given p_b as a lower bound on the rate of the minority class, if $\mathcal{C}$ is PAC-learnable using $\mathcal{H}$ then $\mathcal{C}$ is PAC-learnable with high recall and high precision using $\mathcal{H}$.

- The theorem is true for both realizable and non-realizable learning problems.
- Accomplished by substituting the risk bound ε in the traditional PAC learning framework with $\min\{\varepsilon, \gamma p_b, \xi p_b/2\}$.

How can we compute a lower bound p_b?

Computing a Lower Bound on the Rate of the Minority Class

Algorithm

1. Guess that $p_b = 1/8$.
2. Draw a large enough sample to verify that our guess is correct (whp).
3. If this is true, stop and return p_b, otherwise bisect p_b and go back to the previous step.

Lemma 8

Let $\Pr_{x\sim D}(c(x) = 1) = p > 0$. *Let* $m_i \geq \left\lceil 2^{3+2i} \ln\left(2^{1+i}/\delta\right)\right\rceil$ *for* $i \in \{1, 2, \ldots\}$. *Then, with probability more than* $1-\delta$, *the above algorithm halts within* $\lceil \lg(3/2p) \rceil$ *iterations and provides a lower bound* p_b *such that* $0 < p/8 \leq p_b < p$.

Corollary 9

Lemma 8 requires total sample size $\mathcal{O}\left(\frac{1}{p^2} \cdot \ln\left(\frac{1}{p\delta}\right)\right)$.

(p is the true unknown rate of the minority class.)

The Overhead in the Computation of the Minority CLass

Table: Upper bound on the number of examples requested by our algorithm in order to compute a lower bound (whp) on the rate of the minority class.

Minority Rate (p)	Confidence		
	0.9	***0.95***	***0.99***
20%	13,693	15,356	19,219
10%	61,415	68,069	83,520
5%	272,264	298,881	360,684
1%	8,351,543	9,016,964	10,562,024
0.5%	36,067,831	38,729,516	44,909,758
0.1%	1,056,201,596	1,122,743,726	1,277,249,765
0.05%	4,490,974,869	4,757,143,386	5,375,167,545

PAC Learnability Implies High Recall and High Precision

- Now that we have an algorithm for computing a lower bound p_b on the true rate p of the minority class, we can revisit Theorem 7 and waive the requirement that p_b is given to us ahead of time.

Corollary 10 (of Theorem 7)

$\mathcal{C}$ is PAC-learnable using $\mathcal{H}$ $\Longrightarrow$
$\mathcal{C}$ is PAC-learnable with high recall and high precision using $\mathcal{H}$.

Case Study: Monotone Conjunctions

Monotone Conjunctions/Monomials (Boolean AND of some variables chosen from $\{x_1, x_2, \ldots, x_n\}$)

e.g., $c = x_2 \wedge x_5 \wedge x_8$ (sometimes simply write $c = x_2 x_5 x_8$)

$|\mathcal{H}| = 2^n$ $\text{VC-dim}(\mathcal{H}) = n$

Case Study: Monotone Conjunctions

Monotone Conjunctions/Monomials (Boolean AND of some variables chosen from $\{x_1, x_2, \ldots, x_n\}$)

e.g., $c = x_2 \wedge x_5 \wedge x_8$ (sometimes simply write $c = x_2 x_5 x_8$)

$|\mathcal{H}| = 2^n$ $\text{VC-dim}(\mathcal{H}) = n$

Why use such functions?

- Exhibit **inductive bias**.
- One of the most basic ways of combining features/constraints in a prediction mechanism $\Longrightarrow$ **Explainable/Interpretable functions**.
- Building blocks for richer classes of functions that are less understood; e.g., general DNF formulae.
- Typical benchmarks as they usually provide interesting, but non-trivial insights of the definitions, the bounds that we should expect, etc.
- Can also be useful in contexts of other disciplines.

Setup and Performance Metrics

- Test **two** different **algorithms**: Find-S and the Swapping Algorithm.

$$c = \underbrace{\bigwedge_{i=1}^{m} x_i \wedge \overbrace{\bigwedge_{k=1}^{u} y_k}^{\text{undiscovered}}}_{\text{good}} \qquad \text{and} \qquad h = \overbrace{\bigwedge_{i=1}^{m} x_i}^{\text{mutual}} \wedge \underbrace{\overbrace{\bigwedge_{\ell=1}^{w} z_\ell}^{\text{wrong}}}_{\text{bad}}$$

Proposition 3

Let D be a product distribution over $\{0,1\}^n$ where each variable is satisfied with the same probability λ. Consider a c and an h as above. Then,

$$\begin{cases} R_D(h,c) & = & \lambda^m(\lambda^u + \lambda^w - 2\lambda^{u+w}) \\ \mathrm{Rec}_D(h,c) & = & \lambda^w \\ \mathrm{Prec}_D(h,c) & = & \lambda^u \end{cases}$$

- Uniform distribution obtained for $\lambda = 1/2$. (experiments)

Summary of Experimental Results

Standard PAC learning framework:

- *Both algorithms* may yield prohibitive low recall.
- The Swapping Algorithm in general has better recall, but may have prohibitive low precision, whereas the precision of Find-S is always 1.
  (requiring risk ≤ 0.05, confidence ≥ 0.9.)

Extended PAC learning framework:

- Both Find-S and the Swapping Algorithm identify the target precisely in all the experiments. $\Rightarrow$ Risk 0, Recall 1, Precision 1.
  (requiring risk ≤ 0.05, confidence ≥ 0.9, recall ≥ 0.6, precision ≥ 0.1)

Find-S: Uniform Distribution, PAC Learning

Table: The worst case risk as well as the recall of the generated hypotheses using Find-S under the uniform distribution over 1,000 runs in the traditional PAC framework, with $\varepsilon = 0.05$ and $\delta = 0.1$. Note that the recall of the generated hypotheses can be dramatically low in the traditional PAC framework.

Minority Rate (p)	Max Risk	Recall				Precision
		Min	Median	Mean	Max	
25.0%	0	1	1	1	1	1
12.5%	0	1	1	1	1	1
6.25%	0	1	1	1	1	1
3.125%	0	1	1	1	1	1
1.563%	0	1	1	1	1	1
0.781%	0.781%	$4 \cdot 10^{-10}$	1	0.886	1	1
0.391%	0.391%	$2 \cdot 10^{-28}$	0.25	0.389	1	1
0.195%	0.195%	$4 \cdot 10^{-28}$	$3 \cdot 10^{-5}$	0.078	1	1
0.098%	0.098%	$8 \cdot 10^{-28}$	$2 \cdot 10^{-13}$	0.001	1	1
0.049%	0.049%	$1 \cdot 10^{-27}$	$2 \cdot 10^{-27}$	$2 \cdot 10^{-4}$	0.063	1
0.024%	0.024%	$3 \cdot 10^{-27}$	$3 \cdot 10^{-27}$	$1 \cdot 10^{-5}$	0.008	1

Swapping Algorithm: Uniform Distribution, PAC Learning

Table: The best-case and worst-case risk, the recall and the precision of the generated hypotheses using the Swapping Algorithm under the uniform distribution over 1,000 runs in the traditional PAC framework, with $\varepsilon = 0.05$ and $\delta = 0.1$. Notice that while the recall is better compared to the previous case (Find-S), nevertheless, both the recall and the precision can still be very low compared to what we would like to achieve.

Minority Rate (p)	Risk		Recall				Precision			
	Min	Max	Min	Median	Mean	Max	Min	Median	Mean	Max
25.0%	0	0	1	1	1	1	1	1	1	1
12.5%	0	0	1	1	1	1	1	1	1	1
6.25%	0	0	1	1	1	1	1	1	1	1
3.125%	0	0	1	1	1	1	1	1	1	1
1.563%	1.563%	1.563%	1	1	1	1	50.0%	50.0%	50.0%	50.0%
0.781%	2.344%	3.857%	3.125%	1	70.375%	1	0.781%	25.0%	17.594%	25.0%
0.391%	2.734%	3.491%	3.125%	6.250%	33.494%	1	0.391%	0.781%	4.187%	12.5%
0.195%	2.930%	3.308%	3.125%	3.125%	9.559%	1	0.195%	0.195%	0.597%	6.250%
0.098%	3.027%	3.217%	3.125%	3.125%	5.734%	1	0.098%	0.098%	0.179%	3.125%
0.049%	3.149%	3.171%	3.125%	3.125%	5.216%	25.0%	0.049%	0.049%	0.081%	0.391%
0.024%	3.125%	3.148%	3.125%	3.125%	5.450%	50.0%	0.024%	0.024%	0.043%	0.391%

Experiments in the Extended PAC Learning Framework

Goal 2 (Our Extension of the PAC Learning Framework)

$$\Pr_{S\sim D^m}\left(\begin{array}{lcl} (R_D(h,c) & \leq & \varepsilon) \\ \wedge\ (\mathrm{Rec}_D(h,c) & \geq & 1-\gamma) \\ \wedge\ (\mathrm{Prec}_D(h,c) & \geq & 1-\xi) \end{array}\right) \geq 1-\delta\,.$$

- $\varepsilon = 0.05$, $\delta = 0.1$ (as before). Also use $\gamma = 0.4$, $\xi = 0.9$.
- Find-S generates solutions with precision 1 $\Rightarrow$ Large ξ implies that the value $\min\{\varepsilon, \gamma p_b, \xi p_b/2\}$ (needed by Theorem 7 or Corollary 10) is determined by ε or γp_b.
- Lemma 8 computes a value such that $p/8 \leq p_b < p$.
  $p_b \uparrow \Rightarrow \min\{\varepsilon, \gamma p_b, \xi p_b/2\}$ may increase $\Rightarrow$ the sample size may decrease. So, use $p_b = p$ in the limit in order to make the learning problem as 'hard' as possible. (fewer samples).

<u>**Outcome:**</u> Both Find-S and the Swapping Algorithm identify the target precisely in all the experiments. $\Rightarrow$ Risk 0, Recall 1, Precision 1.

Outline

Summary

1. We extended PAC learning to include explicitly high recall and high precision.
2. We gave lower bounds on the recall and the precision of a learned hypothesis based on its risk and the rate of the minority class.
3. We gave an algorithm to compute a lower bound on the rate of the minority class.
4. $\mathcal{C}$ is PAC learnable $\Rightarrow$ $\mathcal{C}$ is PAC learnable with high recall and high precision.
5. Experimental evaluation by studying two algorithms for learning monotone conjunctions under the uniform distribution.
   (source code: https://github.com/diochnos/pac-imbalanced)

NSF AI Institute for Research on Trustworthy AI in Weather, Climate, and Coastal Oceanography (**AI2ES**)

https://www.ai2es.org **@ai2enviro**

Ideas for Future Work

- Understand better the behavior and the quality of the generated solutions that are obtained by existing PAC algorithms in this new framework.
- Devise new PAC learning algorithms that will have high recall and high precision by design.
- Can we improve the sample size when computing a lower bound on the minority class?
- Connections to other facets of learning; e.g., noise, fairness, ...

Paper: https://doi.org/10.1137/1.9781611976700.4
Supplemental material (omitted discussion and proofs):
http://www.diochnos.com/research/publications/dt-sdm21-supplementary.pdf
Github repository: https://github.com/diochnos/pac-imbalanced

Outline

PAC Learning

Definition 11 (PAC Learning)

A concept class $\mathcal{C}$ is said to be **PAC-learnable** if there exists an algorithm $\mathcal{A}$ and a polynomial function $\text{poly}(\cdot,\cdot,\cdot,\cdot)$ such that for any $\varepsilon > 0$ and $\delta > 0$, for all distributions D on $\mathcal{X}$ and for any target concept $c \in \mathcal{C}$, the following holds for any sample size $m \geq \text{poly}(1/\varepsilon, 1/\delta, n, size(c))$:

$$\Pr_{S\sim D^m}\left(R_D\left(h,c\right) \leq \varepsilon\right) \geq 1-\delta$$

If $\mathcal{A}$ further runs in $\text{poly}(1/\varepsilon, 1/\delta, n, size(c))$, then $\mathcal{C}$ is said to be **efficiently PAC-learnable**. When such an algorithm $\mathcal{A}$ exists, it is called a **PAC-learning algorithm** for $\mathcal{C}$.

- $size(c)$ denotes the maximal cost for the representation of $c \in \mathcal{C}$.
  Example: Representing a monotone conjunction as a list of the k variables that pose the constraints, takes space $\mathcal{O}\left(k \log n\right)$.

Agnostic PAC Learning

Definition 12 (Agnostic PAC Learning)

Let $\mathcal{H}$ be a hypothesis space. Algorithm $\mathcal{A}$ is an **agnostic PAC-learning algorithm** if there exists a polynomial function $\text{poly}(\cdot,\cdot,\cdot,\cdot)$ such that for any $\varepsilon > 0$, $\delta > 0$, for all distributions D over $\mathcal{X} \times \mathcal{Y}$, the following holds for any sample size $m \geq \text{poly}(1/\varepsilon, 1/\delta, n, size(c))$:

$$\Pr_{S\sim D^m}\left(R_D\left(h,c\right) \leq \min_{h^\star \in \mathcal{H}} \left\{R_D\left(h^\star,c\right)\right\} + \varepsilon\right) \geq 1-\delta$$

If $\mathcal{A}$ further runs in $\text{poly}(1/\varepsilon, 1/\delta, n, size(c))$, then it is said to be an **efficient agnostic PAC-learning algorithm**.

Remark 1

We have a more general scenario (stochastic) since D is defined on $\mathcal{X} \times \mathcal{Y}$. (The label of the point is not unique.)

PAC Learning Extension

Definition 13 (PAC Learning Extension)

A concept class $\mathcal{C}$ is said to be **PAC-learnable with high recall and high precision** by a hypothesis space $\mathcal{H}$, if there exists a learning algorithm $\mathcal{A}$ and a polynomial function $poly(\cdot,\cdot,\cdot,\cdot,\cdot,\cdot)$, such that for any $\varepsilon > 0$, $\delta > 0$, $\gamma > 0$, and $\xi > 0$, for all distributions $D \in \mathcal{D}$ over $\mathcal{X}$, for any target concept $c \in \mathcal{C}$, for any sample $\mathcal{S}$ of size $m \geq poly(1/\epsilon, 1/\delta, 1/\gamma, 1/\xi, n, size(c))$, algorithm $\mathcal{A}$ outputs a hypothesis $h \in \mathcal{H}$, such that:

$$\Pr_{S\sim D^m}\left(\begin{array}{rcl} (R_D(h,c) & \leq & \varepsilon) \\ \wedge\ (\mathrm{Rec}_D(h,c) & \geq & 1-\gamma) \\ \wedge\ (\mathrm{Prec}_D(h,c) & \geq & 1-\xi) \end{array}\right) \geq 1-\delta$$

Furthermore, if $\mathcal{A}$ runs in time $poly(1/\epsilon, 1/\delta, 1/\gamma, 1/\xi, n, size(c))$, then $\mathcal{C}$ is said to be **efficiently PAC-learnable with high recall and high precision** by the hypothesis space $\mathcal{H}$.

The Vapnik-Chervonenkis Dimension

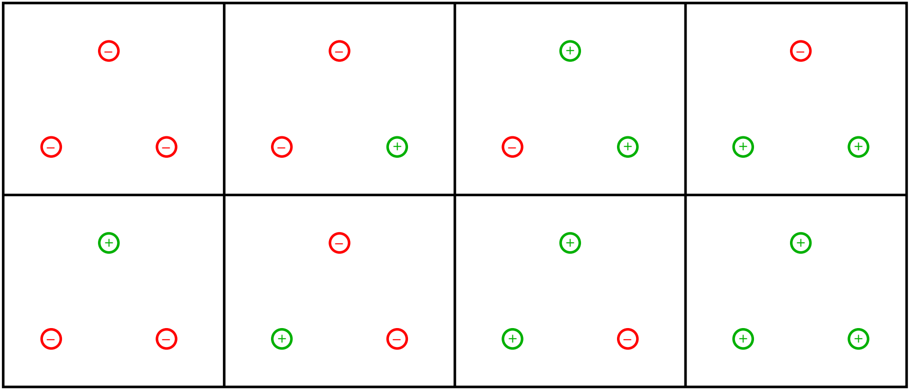
Definition 14

A set of instances $\{x_1, \ldots, x_d\}$ is *shattered* by $\mathcal{H}$, if for every possible labeling $y_1, \ldots, y_d$, there exists an $h \in \mathcal{H}$ such that $h(x_i) = y_i$ for every $i \in \{1, \ldots, d\}$. That is, there are 2^d distinct classifications of the instances $\{x_1, \ldots, x_d\}$ that can be realized by hypotheses in $\mathcal{H}$.

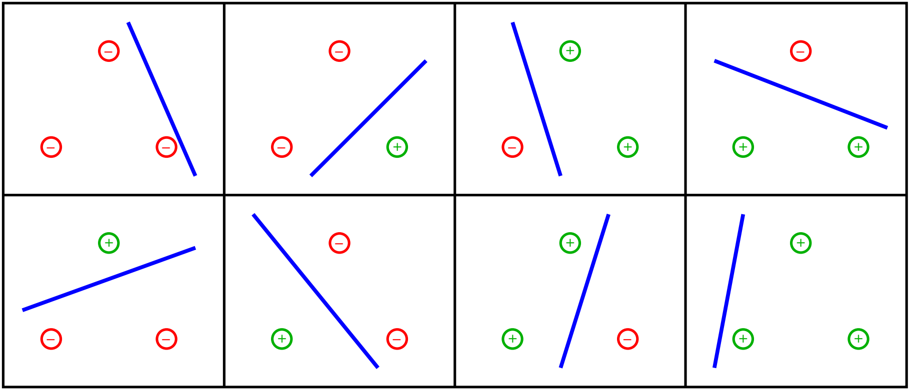
Definition 15 (*VC* dimension)

The Vapnik-Chervonenkis dimension (or VC dimension) of $\mathcal{H}$ is defined as the largest integer d for which there exists a set of instances $\{x_1, \ldots, x_d\}$ that is shattered by $\mathcal{H}$.

Configurations of 3 Points in 2D



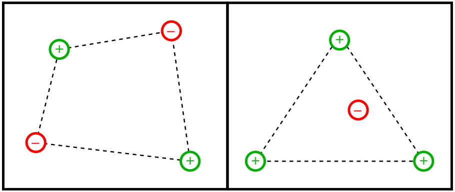
Halfspaces Shatter 3 Points in 2D



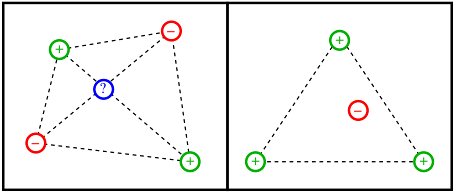
Question 1

Can we shatter 4 points ?

Can Halfspaces Shatter 4 Points in 2D?



Halfspaces *cannot* Shatter 4 Points in 2D



Theorem 16 (Radon)

Any set of $d + 2$ points in R^d can be partitioned into two (disjoint) sets whose convex hulls intersect.

Corollary 17

- $\textit{VC-dim}(\textit{HALFSPACES}) = 3$ *in* 2 *dimensions.*
- $\textit{VC-dim}(\textit{HALFSPACES}) = d + 1$ *in* $d \geq 1$ *dimensions.*

The Algorithm Find-S

1. Initialize the hypothesis to be the full conjunction of all the variables.
2. Request m examples (per a PAC bound) and look at the positive ones.
3. Delete the variables that are falsified in the positive examples.

$(\mathcal{H} = \mathcal{C} \Rightarrow$ *proper* learning $\Rightarrow$ *realizable* case$)$

A Study of Thinking [Bruner, Goodnow, Austin, 1956], *Machine Learning* [Mitchell, 1997]

Example 1

Let $\mathcal{X} = \{0,1\}^{10}$ and $c = x_2 \wedge x_4 \wedge x_5$.

example	hypothesis h
	$x_1 \wedge x_2 \wedge x_3 \wedge x_4 \wedge x_5 \wedge x_6 \wedge x_7 \wedge x_8 \wedge x_9 \wedge x_{10}$
$((1101111101), +)$	$x_1 \wedge x_2 \wedge x_4 \wedge x_5 \wedge x_6 \wedge x_7 \wedge x_8 \wedge x_{10}$
$((0101111101), +)$	$x_2 \wedge x_4 \wedge x_5 \wedge x_6 \wedge x_7 \wedge x_8 \wedge x_{10}$
$((1101110111), +)$	$x_2 \wedge x_4 \wedge x_5 \wedge x_6 \wedge x_8 \wedge x_{10}$
$((0101110100), +)$	$x_2 \wedge x_4 \wedge x_5 \wedge x_6 \wedge x_8$

The Swapping Algorithm (on Monotone Conjunctions)

- Local search method; the **neighborhood** is defined by:
  - **Adding**, **removing**, or **swapping** a variable in the hypothesis.
- **The learner cannot see individual training examples**, but instead, based on a sample S receives as input the value

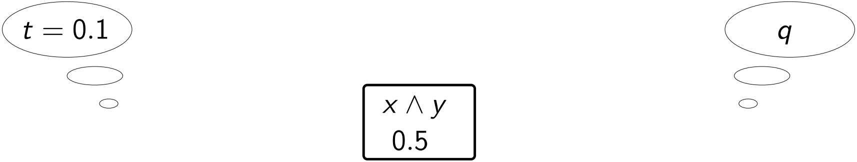
$$\mathrm{Perf}_D\left(h, c, S\right) = \frac{1}{|S|}\sum_{x \in S} h(x)c(x)\,.$$

- This is an **approximation of the true correlation** that the hypothesis h has with the target c:

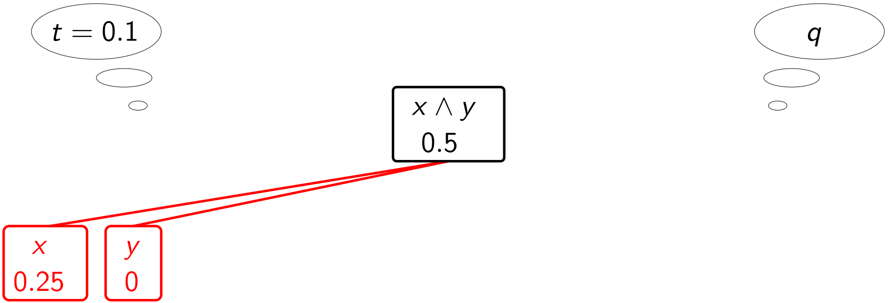
$$\mathrm{Perf}_D\left(h, c\right) = \sum_{x \in \mathcal{X}} h(x)c(x)D(x) = 1 - 2 \cdot \Pr\left(h(x) \neq c(x)\right)$$

- Using a threshold $\boldsymbol{t}$ the neighborhood is partitioned into three parts: Beneficial, Neutral, and Deleterious.
- Then the learner selects a hypothesis at random from the most promising non-empty set.

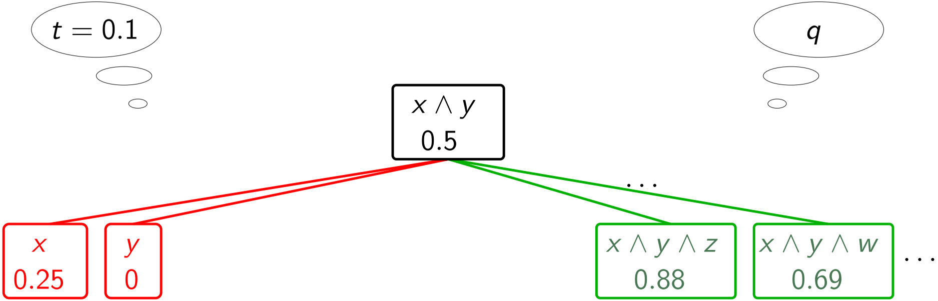
The Swapping Algorithm



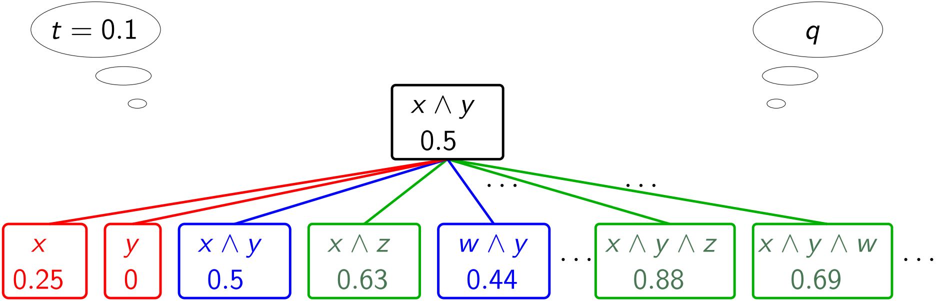
The Swapping Algorithm



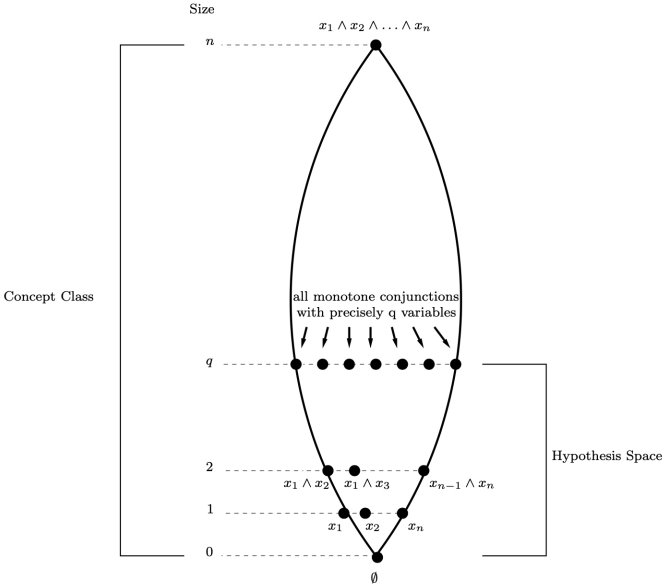
The Swapping Algorithm



The Swapping Algorithm



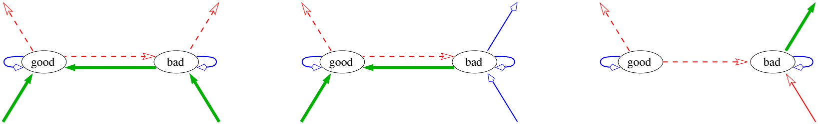
The Hypothesis Space for the Swapping Algorithm



Convergence of the Swapping Algorithm

- Uniform distribution over $\{0, 1\}^n$ [Valiant, 2009], [D & Turán, 2009]
- Product distributions where all variables are satisfied with the same probability λ [D, 2016]

Example 1: Short Initial Hypothesis and Short Target



(a) $u \geq 2$ (b) $u = 1$ (c) $u = 0$

Let $\mathcal{X}_8 = \{0,1\}^8$ such that $\{g_1, g_2, g_3, b_1, b_2, b_3, b_4, b_5\}$, the target be $c = g_1 \wedge g_2 \wedge g_3$, and require $\varepsilon = 1/5$. $(q = 4)$

Step i	u	Hypothesis h_i	Performance	Neighborhood	Class
0		$\emptyset$	$-3/4$	N^+	
1		b_1	0	$N^+ \cup \{\text{swaps: } b \to g\}$	
2	≥ 2	$b_1 \wedge b_2$	$3/8$	$N^+ \cup \{\text{swaps: } b \to g\}$	
3		$b_1 \wedge b_2 \wedge b_3$	$9/16$	$N^+ \cup \{\text{swaps: } b \to g\}$	
4		$b_1 \wedge b_2 \wedge b_3 \wedge b_4$	$21/32$	$\{\text{swaps: } b \to g\}$	Bene
5		$b_1 \wedge g_3 \wedge b_3 \wedge b_4$	$22/32$	$\{\text{swaps: } b \to g\}$	
6	1	$g_1 \wedge g_3 \wedge b_3 \wedge b_4$	$24/32$	$\{\text{swaps: } b \to g\}$	
7	0	$g_1 \wedge g_3 \wedge g_2 \wedge b_4$	$28/32$	$\{\text{remove } b\}$	
8	0	$g_1 \wedge g_3 \wedge g_2$	1	$\{h_8\}$	Neut

Example 2: Short Initial Hypothesis and Long Target

Let $\mathcal{X}_{13} = \{0,1\}^{13}$ such that $\{g_1, g_2, g_3, g_4, g_5, g_6, g_7, b_1, b_2, b_3, b_4, b_5, b_6\}$, the target be $c = g_1 \wedge g_2 \wedge g_3 \wedge g_4 \wedge g_5 \wedge g_6 \wedge g_7$, and require $\varepsilon = 1/5$. $(q = 4)$

Step i	u	Hypothesis h_i	Performance	Neighborhood	Class
0		$\emptyset$	$-63/64$	N^+	
1	≥ 2	b_1	0	$N^+ \cup \{\text{swaps: } b \to g\}$	Bene
2		$b_1 \wedge b_2$	$63/128$	$N^+ \cup \{\text{swaps: } b \to g\}$	
3		$b_1 \wedge b_2 \wedge b_3$	$189/256$	$N^+ \cup \{\text{swaps: } b \to g\}$	
4		$b_1 \wedge b_2 \wedge b_3 \wedge b_4$	$425/512$	$\{\text{all swaps}\} \cup \{h_4\}$	
5	≥ 2	$b_1 \wedge b_6 \wedge b_3 \wedge b_4$	$425/512$	$\{\text{all swaps}\} \cup \{h_5\}$	Neut
6		$b_1 \wedge b_6 \wedge b_3 \wedge b_5$	$425/512$	$\{\text{all swaps}\} \cup \{h_6\}$	
7		$b_1 \wedge b_6 \wedge b_3 \wedge b_5$	$425/512$	$\{\text{all swaps}\} \cup \{h_7\}$	
8		$g_1 \wedge b_6 \wedge b_3 \wedge b_5$	$426/512$	$\{\text{swaps: } b \to g\}$	
9	≥ 2	$g_1 \wedge b_6 \wedge b_3 \wedge g_4$	$428/512$	$\{\text{swaps: } b \to g\}$	Bene
10		$g_1 \wedge b_6 \wedge g_6 \wedge g_4$	$432/512$	$\{\text{swaps: } b \to g\}$	
11		$g_1 \wedge g_3 \wedge g_6 \wedge g_4$	$440/512$	$\{\text{swaps: } g \to g\} \cup \{h_{11}\}$	
12	≥ 2	$g_1 \wedge g_3 \wedge g_5 \wedge g_4$	$440/512$	$\{\text{swaps: } g \to g\} \cup \{h_{12}\}$	Neut
13		$g_1 \wedge g_3 \wedge g_5 \wedge g_4$	$440/512$	$\{\text{swaps: } g \to g\} \cup \{h_{13}\}$	
14		$g_2 \wedge g_3 \wedge g_5 \wedge g_4$	$440/512$	$\{\text{swaps: } g \to g\} \cup \{h_{14}\}$	