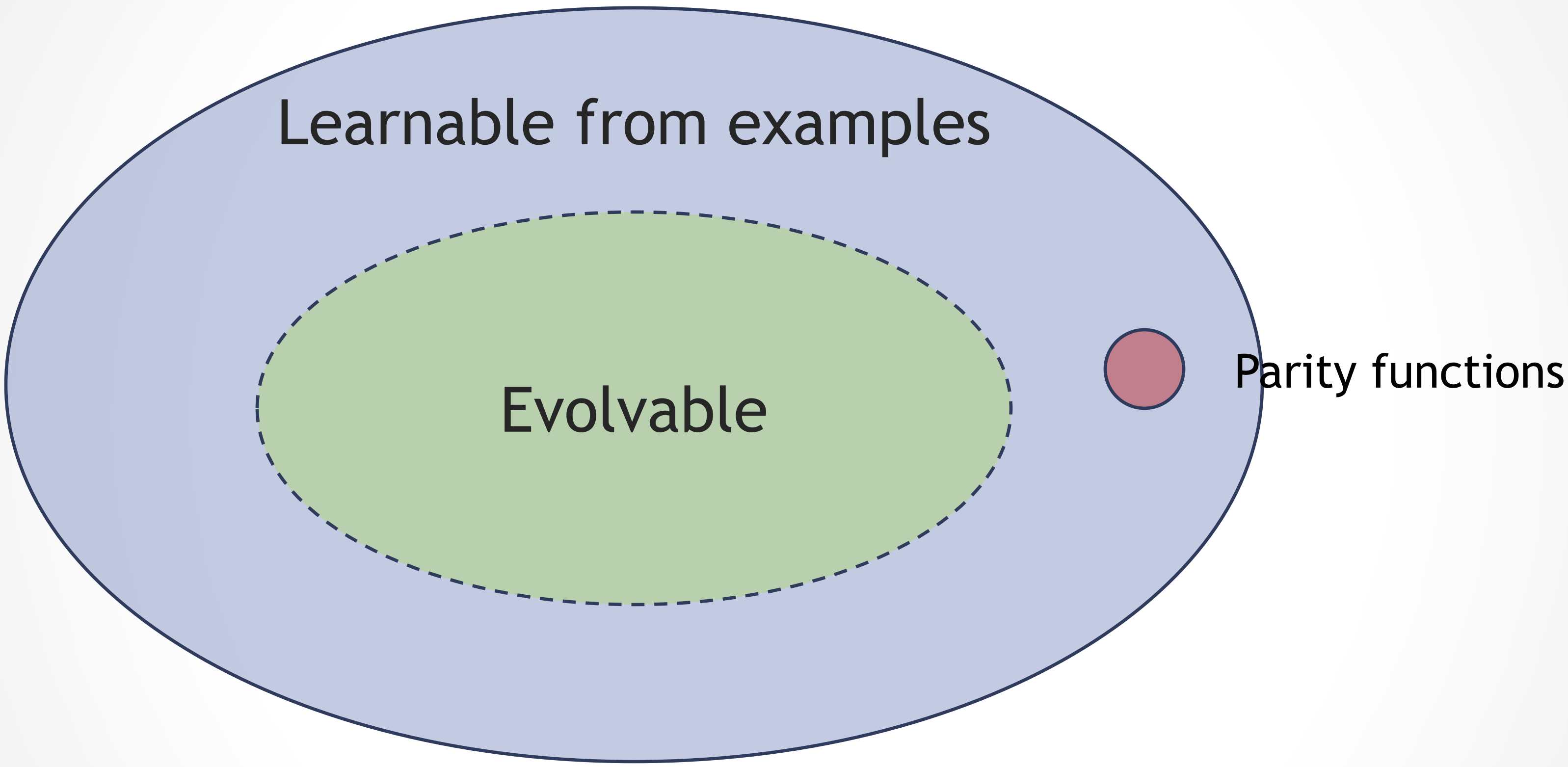
On the power and the limits of evolvability

Vitaly Feldman
Almaden Research Center

Learning from examples vs evolvability



The core model: PAC [Valiant 84]

- Learner observes random examples: $(x, f(x))$
- Assumption: unknown Boolean function $f: X \to \{-1,1\}$ labeling the examples comes from a known class of functions C
- Distribution D over X (e.g. R^n or $\{-1,1\}^n$)

every distribution D

For every $f \in C, \epsilon > 0$, w.h.p. output $h: X \to [-1,1]$

s.t. $\mathop{\mathrm{E}}_{x \sim D}[f(x)h(x)] \geq 1 - \epsilon$

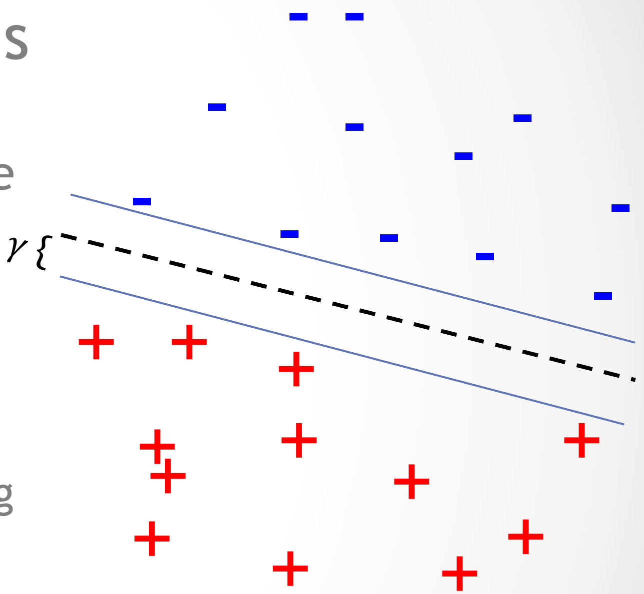
For Boolean h

$\Pr_{x \sim D}[f(x) = h(x)] \geq 1 - \epsilon/2$

Efficient: poly($\frac{1}{\epsilon}$, $|x|$) time

Classical example

- C halfspaces/linear threshold functions
  - $sign(\sum_i w_i x_i \ - \theta)$ for $w_1, w_2, \ldots, w_n, \theta \in \boldsymbol{R}$
  - equivalent to examples being linearly separable



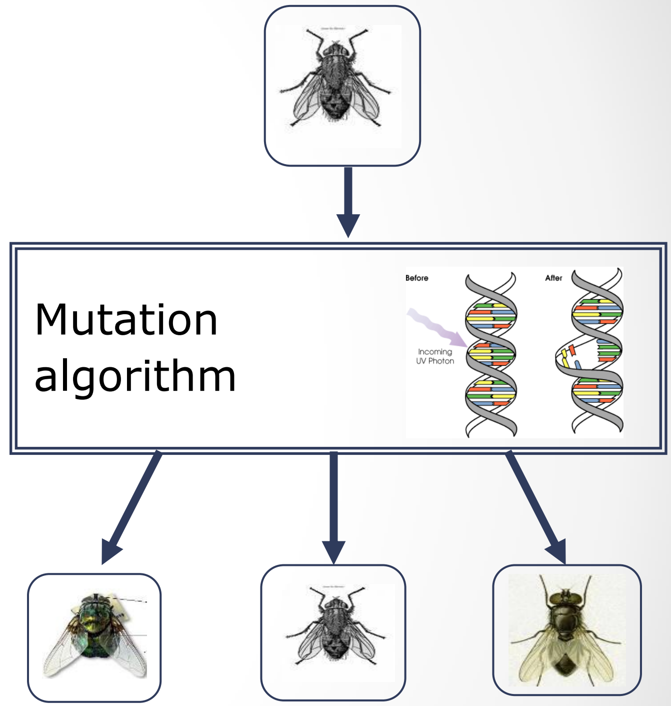
- Perceptron algorithm

  [Rosenblatt 57; Block 62; Novikoff 62]
  - Start with LTF h^0 defined by $w^0 = (0,0,\ldots,0); \theta^0 = 0$
  - Get a random example (x, ℓ). If $h^t(x) = \ell$ do nothing
  - Else let h^{t+1} be LTF defined by

    $$w^{t+1} = w^t + \ell \cdot x \,; \theta^{t+1} = \theta^t + \ell$$
- Gives PAC learning if f has significant margin γ on the observed data points

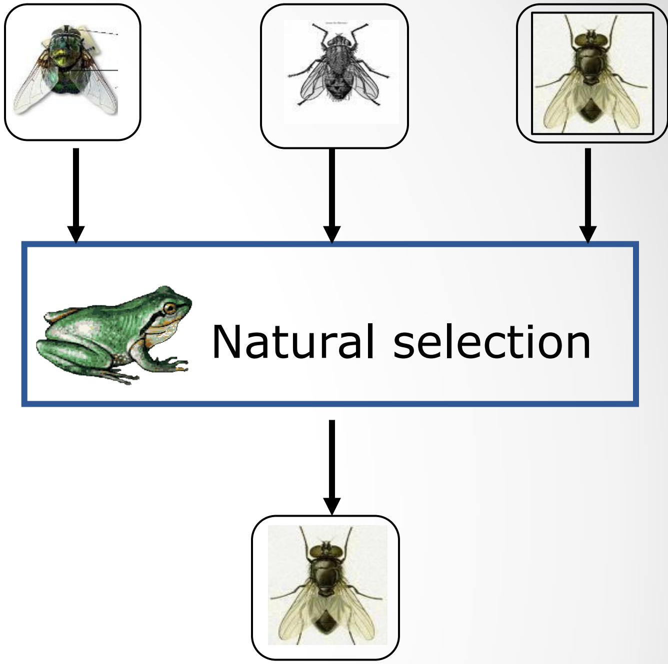
Evolution algorithm

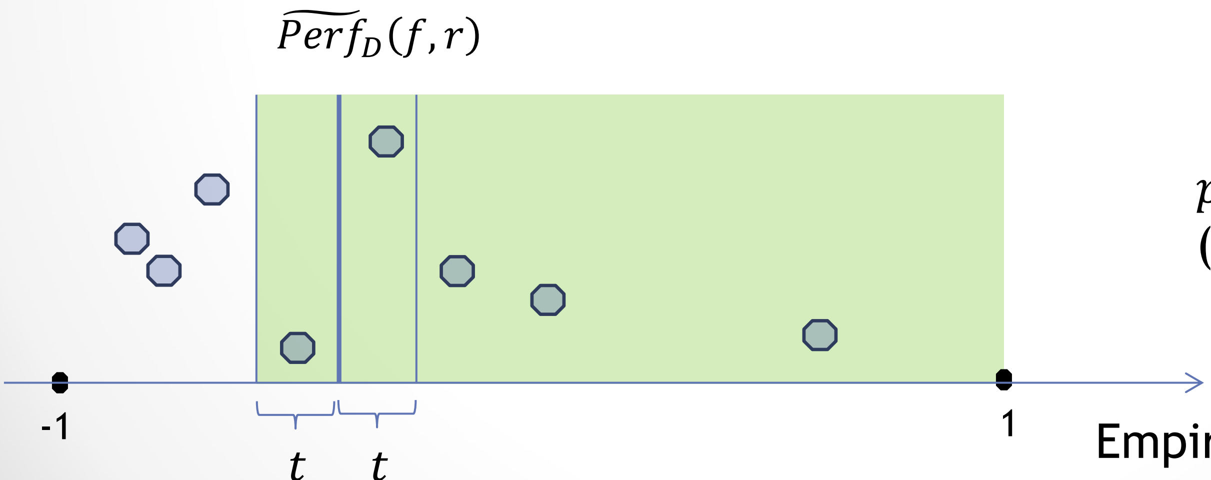
- R - representation class of functions over X
  - E.g. all linear thresholds over $\boldsymbol{R}^n$
- M - randomized *mutation algorithm* that given $r \in R$ outputs (a random mutation) $r' \in R$
  - Efficient: poly in $\frac{1}{\epsilon}, n$
  - E.g. choose a random i and adjust w_i by 0, +1 or -1 randomly



Selection

- Fitness $Perf_D(f,r) \in [-1,1]$
  - Correlation: $E_D[f(x)r(x)]$
- For $r \in R$ sample $M(r)$ p times: $r_1, r_2, \dots, r_p$
- Estimate empirical performance of r and each r_i using s samples: $\widetilde{Perf}_D(f, r_i)$





p and s are "feasible" (polynomial in $n, 1/\epsilon$)

Evolvability

- Class of functions C is evolvable over D if exists an evolution algorithm (R, M) and a polynomial $g(\cdot,\cdot)$ s.t.

For every $f \in C, r \in R, \varepsilon > 0$ and a sequence $r_0 = r, r_1, r_2, \ldots$ where $r_{i+1} \leftarrow \text{Select}(R, M, r_i)$ it holds: $Perf_D(f, r_{g\left(n, \frac{1}{\epsilon}\right)}) \geq 1 - \varepsilon$ w.h.p.

- Evolvable (*distribution-independently)*
  - Evolvable for all D by the same mutation algorithm (R, M)

Limits of evolvability

- Feedback is restricted to values of $\widetilde{Perf}_D(f, r_i)$ for some polynomial number of samples s

learning algorithm

$\xrightarrow{\quad h_1 \quad}$

$\xleftarrow{\quad v_1 \quad}$

$\xrightarrow{\quad h_2 \quad}$

$\xleftarrow{\quad v_2 \quad}$

$\vdots$

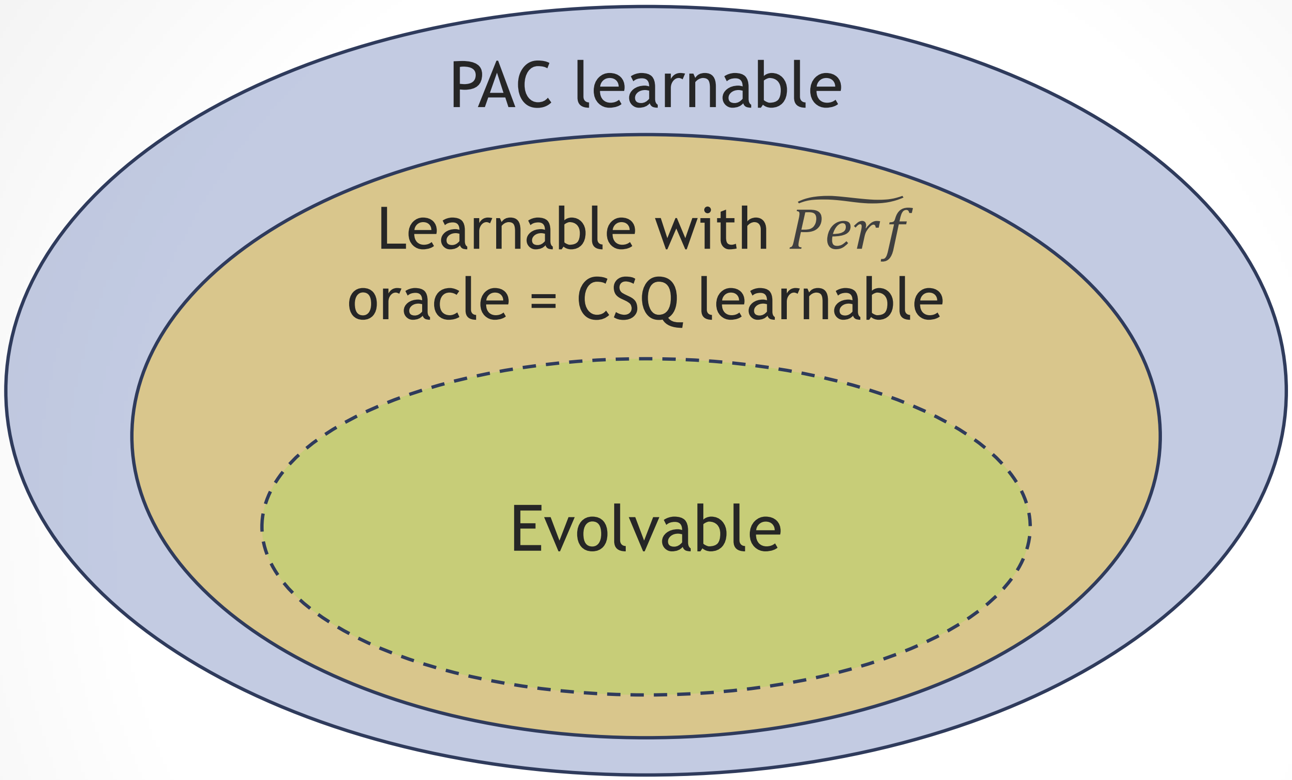
$\xrightarrow{\quad h_q \quad}$

$\xleftarrow{\quad v_q \quad}$

$\widetilde{Perf}_D(s)$ oracle

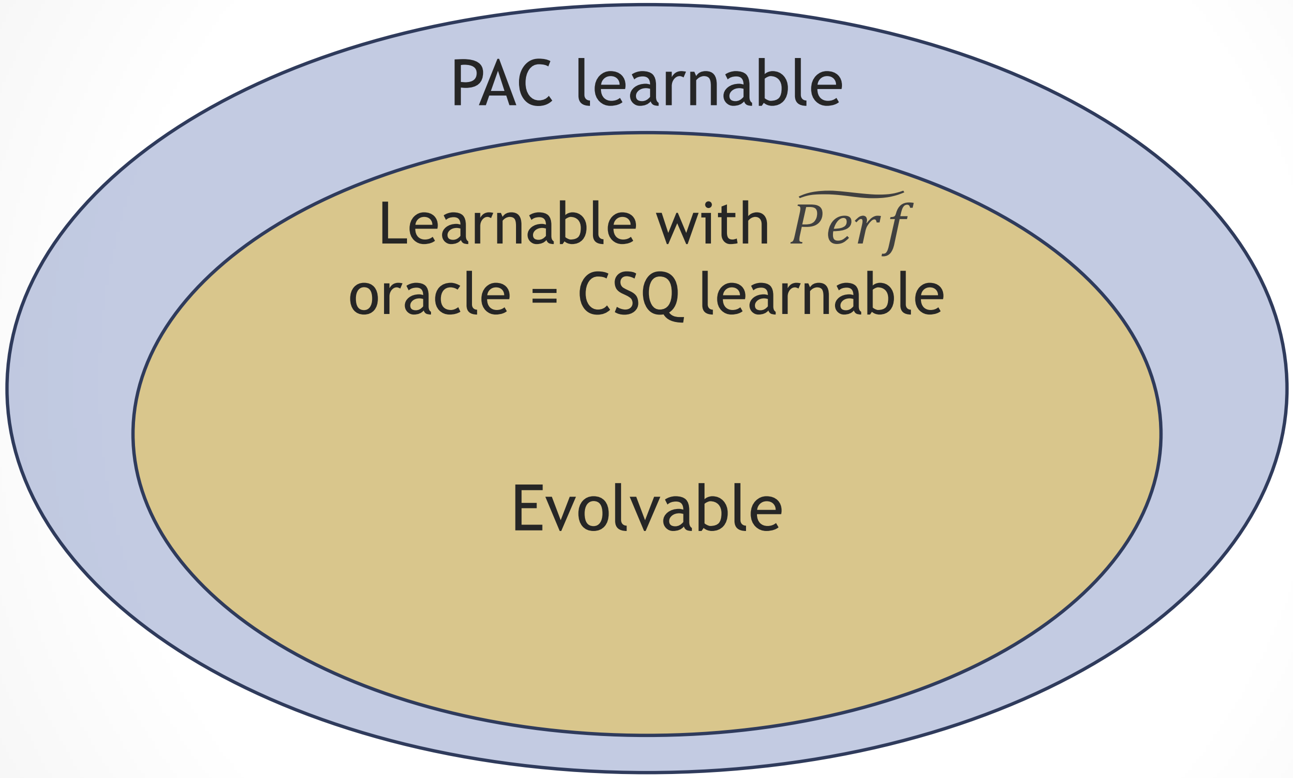
$v_i = \widetilde{Perf}_D(f, h_i)$ evaluated on s fresh examples

Evolvable ⊆ CSQ learnable



Correlational Statistical Query:
To query h CSQ oracle responds with any value v
$|v - \boldsymbol{E}_D[f(x)h(x)]| \leq \tau$ for $\tau \geq \frac{1}{poly\left(n,\frac{1}{\epsilon}\right)}$
Learning by Distances [Ben-David,Itai,Kushilevitz '90]
Restriction of SQ model by Kearns [93]

CSQ learnable ⊆ Evolvable [F. 08]



Proof outline

Replace queries for performance values with approximate comparisons

For hypothesis $h: X \to [-1,1]$, tolerance $\tau > 0$ and threshold $t \geq \tau$, $\mathrm{CSQ}_{>}$ oracle returns:

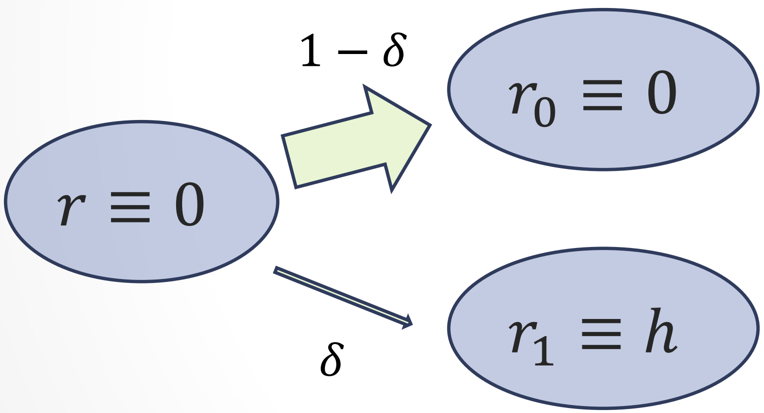
1	if $\boldsymbol{E}_D[f(x)h(x)] \geq t + \tau$
0	if $\boldsymbol{E}_D[f(x)h(x)] \leq t - \tau$
0 or 1	otherwise

Design evolution algorithm with mutations that simulate comparison queries

From comparisons to mutations

For hypothesis $h: X \to [-1,1]$, tolerance $\tau > 0$ and threshold $t \geq \tau$, $\mathrm{CSQ}_{>}$ oracle returns:

$$\begin{cases} 1 & \text{if } \boldsymbol{E}_D[f(x)h(x)] \geq t + \tau \\ 0 & \text{if } \boldsymbol{E}_D[f(x)h(x)] \leq t - \tau \\ 0 \text{ or } 1 & \text{otherwise} \end{cases}$$



Beneficial/neutral threshold = t

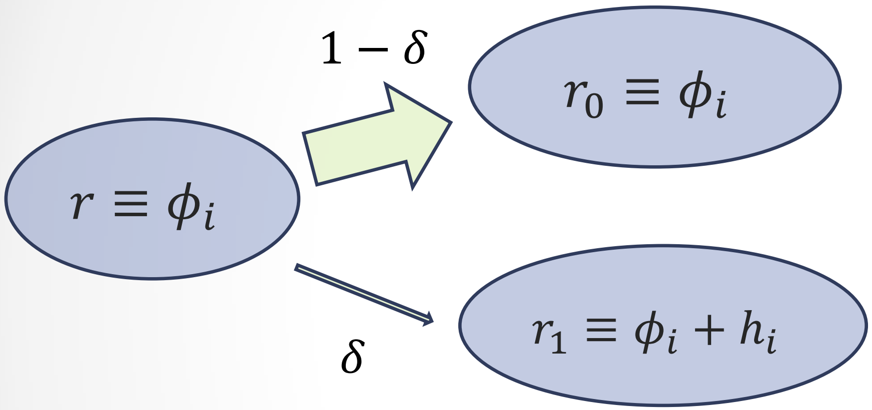
Mutation pool size $p = O\left(\frac{\log(\frac{1}{\delta})}{\delta}\right)$

Performance sample size $s = O\left(\frac{\log(\frac{1}{\delta})}{\tau^2}\right)$

With probability at least $1 - \delta$, Select$(r) = r_b$ where b is valid response to comparison query

Simulating $CSQ_>$ algorithm

Need to answer q queries
ϕ_i is the function obtained after answering i queries
Need to answer query h_i with threshold t_i



Beneficial/neutral threshold = t_i

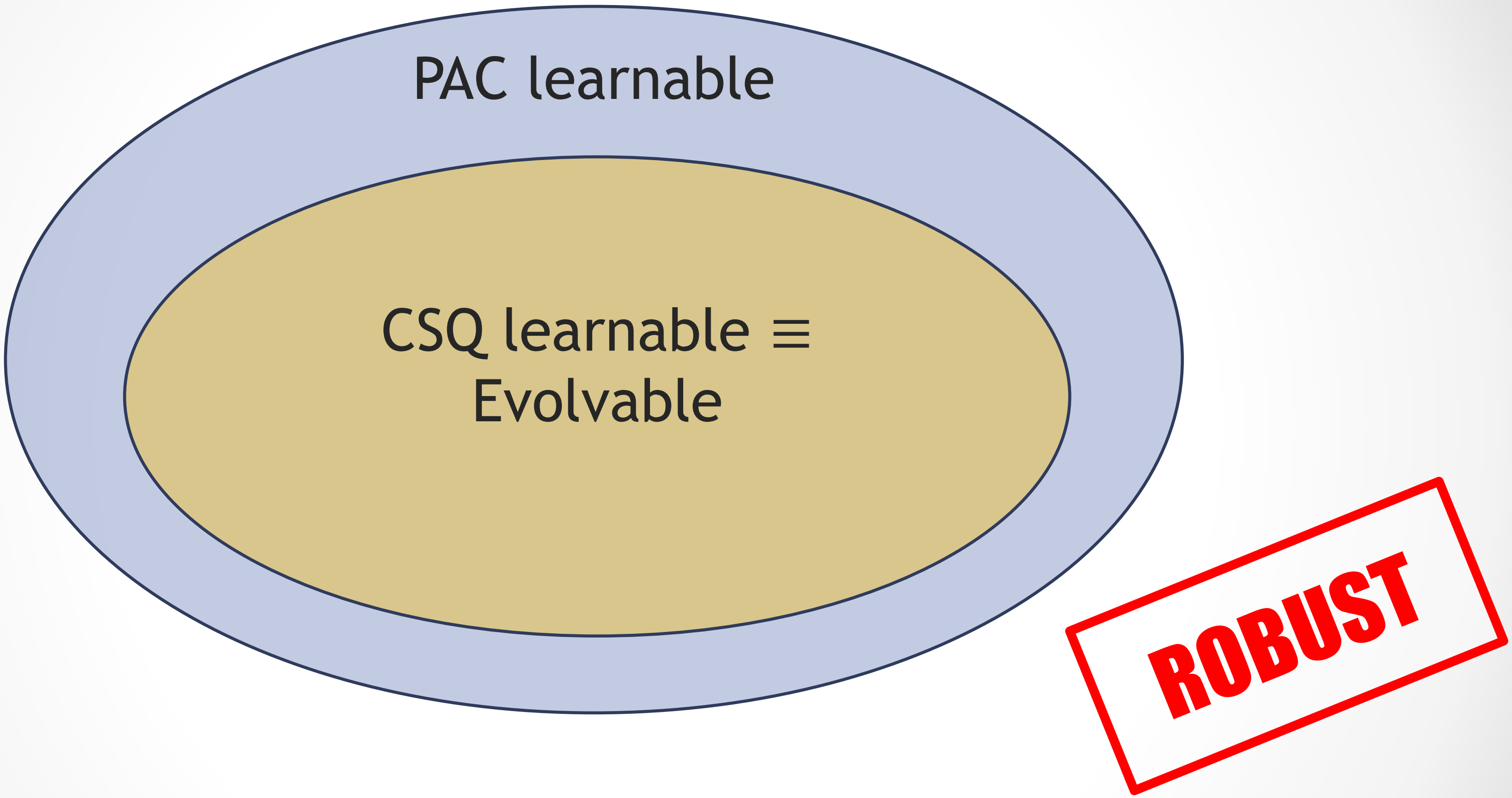
Mutation sample size $p = O\left(\frac{\log(\frac{1}{\delta})}{\delta}\right)$

Performance sample size $s = O\left(\frac{\log(\frac{1}{\delta})}{\tau^2}\right)$

Leads to representations with values in $[-q, q]$!
Rescale by $1/q$ to get functions with values in [-1,1]

Given answers to queries can compute h such that $Perf_D(f, h) \geq 1 - \epsilon$ and mutate into it

CSQ learnable ≡ Evolvable [F.08; F. 09]



E.g. optimizing selection; recombination [Kanade 11]; changing thresholds; number of mutations

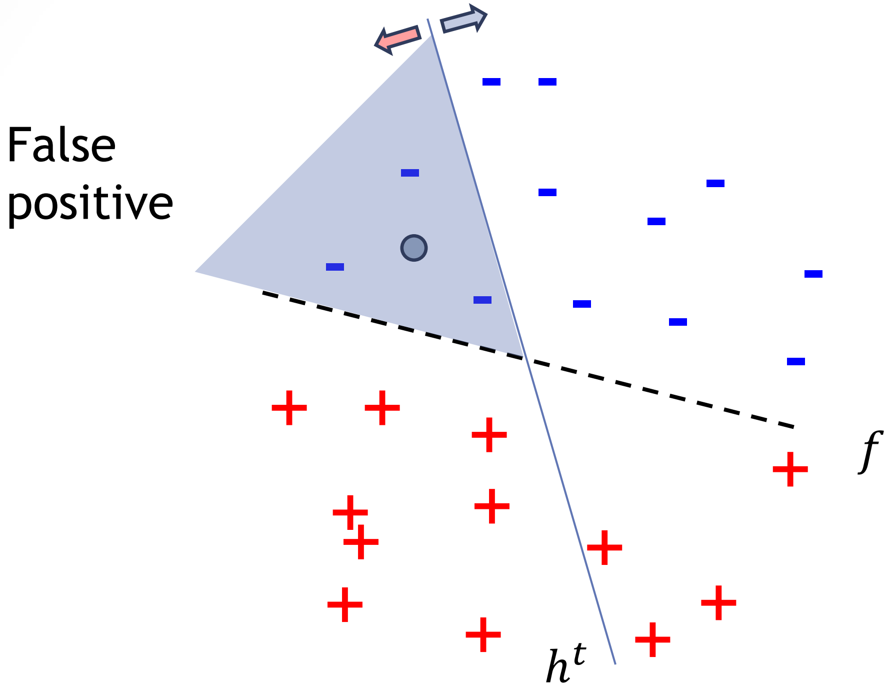
How powerful is CSQ learning?

SQ learning [Kearns 93] : learner submits $\psi(x,\ell)$
SQ oracle returns v such that $|v - \boldsymbol{E}_D[\psi(x, f(x))]| \leq \tau$

- Many known algorithms are essentially described in SQ or can be easily translated to SQ
  - Boolean conjunctions, decision lists, simple geometric concepts, AC^0 [Kearns 93]
- Several other were found with more effort
  - Halfspaces with large margin [Bylander 94]
  - General LTFs [BlumFrie.Kann.Vemp. 96; Duna.Vemp. 04]
- General ML techniques
  - Nearest neighbor
  - Gradient descent
  - SVM
  - Boosting

Perceptron in SQ

- Recall the Perceptron algorithm:
  - Add false negatives, subtract false positive examples



- Use SQs to find the centroid of false positives
- $\boldsymbol{E}_D[x_1 \cdot I(f(x) = -1) \cdot I(h^t(x) = 1)]$ gives the first coordinate of the centroid
- Use the centroid for the Perceptron update
-

If D is fixed then SQ≡CSQ

- Decompose SQ into CSQ and a constant

$$\begin{aligned}\mathbf{E}_D[\psi(x, f(x))] &= \mathbf{E}_D\left[\psi(x,-1)\frac{1-f(x)}{2} + \psi(x,1)\frac{1+f(x)}{2}\right] \\ &= \mathbf{E}_D\left[\frac{\psi(x,1)-\psi(x,-1)}{2}f(x)\right] + \mathbf{E}_D\left[\frac{\psi(x,1)+\psi(x,-1)}{2}\right]\end{aligned}$$

CSQ

- Corollary: linear threshold functions are evolvable for any fixed distribution D

Distribution-independent CSQ

- Single points are learnable [F. 09]
- Characterization of weak-learning [F. 08]

Better than random guessing: $\mathop{\mathrm{E}}_{x\sim D}[f(x)h(x)] \geq \frac{1}{poly\left(n,\frac{1}{\epsilon}\right)}$

C is weakly CSQ learnable if and only if all functions in C can be represented as linear threshold functions with "significant" margin over a poly-size set of Boolean features

- General linear thresholds are not weakly CSQ learnable [Goldmann,Hastad,Razborov 95] (but are SQ learnable)
- Conjunctions are not CSQ learnable [F. 11]

Further directions

- Characterize (strong) evolvability (CSQ learning)
    - Strengthen the lower bound for conjunctions
- Are thresholds on a line evolvable distribution independently
- $Perf_D(f,r) = -\boldsymbol{E}_D\left[\left(f(x)-r(x)\right)^2\right]$ then all of SQ is evolvable [F. 09]