A Comparison of Antenna Placement Algorithms

Abhinav Jauhri

References

- *Evolvability*, Valiant (2009)
- *A Complete Characterization of Statistical Query Learning with Applications to Evolvability*, Feldman (2009)
- Talk by Feldman on *Distribution-Independent Evolvability of Linear Threshold Functions*, COLT 2011
- Talk on *Quantititative Model of Innovation in Evolution* by Valiat (2014)

Antenna Placement Objectives

Antenna Placement Issues

- → Coupling among antennas
- → Parasitic effects and reflections from the host platform
- → Difficulty conforming to aerodynamic, thermal, other enovironment factors

Desired Antenna Placement Objectives

- → Gain in radiation pattern
- → Minimize coupling
- → Pattern shape objectives in azimuth and/or elevation

Example

Expression of a gene in the DNA regulated by *transcription factors* (TF). They look at the environment factors; other proteins and then determine how much protein will be produced by the gene.

Example

Expression of a gene in the DNA regulated by *transcription factors* (TF). They look at the environment factors; other proteins and then determine how much protein will be produced by the gene.

Then, the question would be - how do regulations mechanisms evolve?

Evolution Algorithm

- R - representation class of functions over domain X.
  Example: all linear thresholds over $\mathbb{R}^n$.

- M - randomized mutation algorithm that given $r \in R$ outputs $r' \in R$
  - Efficient: poly in $\frac{1}{\epsilon}, n$

Selection

- Fitness: $\mathrm{Perf}_D(f, r) \in [-1, 1]$, where f is some ideal function; r function computed by our representation
  - Correlation $E_D[f(x)r(x)]$*
- Run $M(r)$ p times to get $R = \{r_1, r_2, \ldots r_p\}$
- Estimate $\forall r_i \in R$, $\widetilde{\mathrm{Perf}}_D(f, r_i) = \frac{1}{S} \sum\limits_{j \leqslant S} r_i(x_j) f(x_j)$

p and S should be poly in n, $\frac{1}{\epsilon}$

*CSQ is a special case of SQ model

Evolvability (formally)

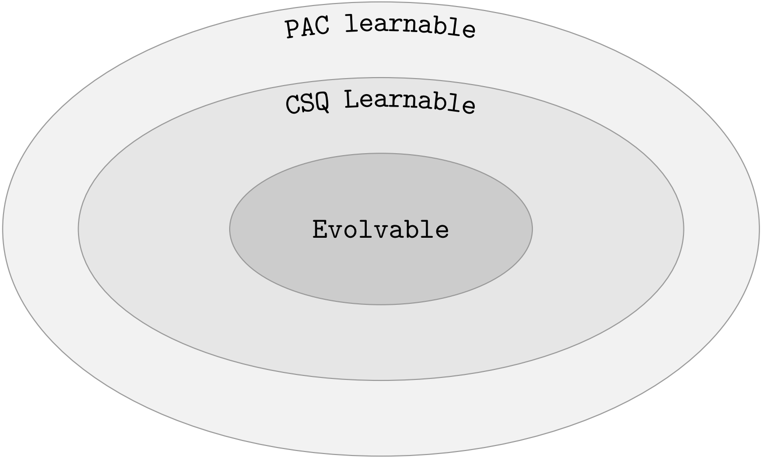
Class of functions C is evolvable over D if exists an evolution algorithm (R, M) and a polynomial g s.t.

> for every $f \in C, r \in R, \epsilon > 0$ and a sequence $r_0 = r, r_1, \ldots$ where $r_{i+1} \leftarrow \mathit{select}(R, M, r_i)$, it holds $\mathrm{Perf}_D(f, r_{g(n, \frac{1}{\epsilon})}) \geqslant 1 - \epsilon$ w.h.p.

Evolvability Oracle

- Send oracle h_1
- Oracle returns $v_1 = \widetilde{\mathrm{Perf}}_D(f, h_1)$ by using S fresh examples
- Send oracle h_2
- Oracle returns $v_2 = \widetilde{\mathrm{Perf}}_D(f, h_2)$ by using S fresh examples
- ...
- ...

Big Picture

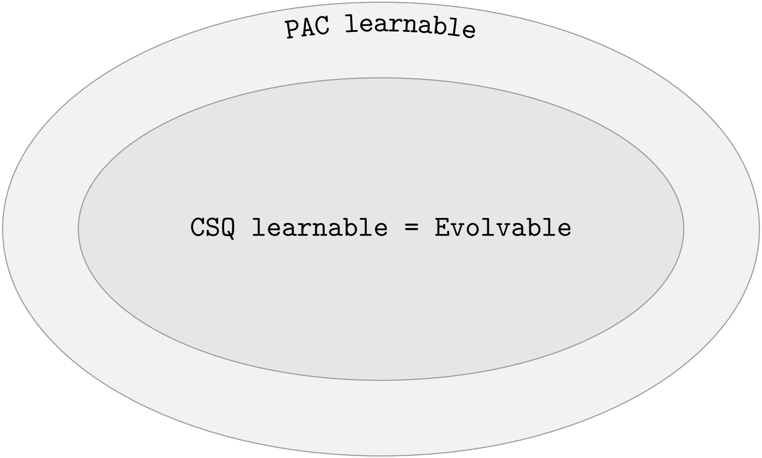


Equivalence with CSQ

Correlational Statistical Query:

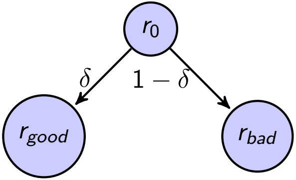
- Send oracle query h

- CSQ oracle responds with $|v - \widetilde{\text{Perf}}_D(f, h)| \leqslant \tau$, and $\tau \geqslant \frac{1}{poly(n, \frac{1}{\epsilon})}$ (to ensure tolerance isn't very small)


PAC learnable
CSQ learnable = Evolvable

Proof Outline

for some $\tau > 0$, and $t \geqslant \tau$ CSQ oracle returns:
1 if $E_D[f(x)h(x)] \geqslant t + \tau$
0 if $E_D[f(x)h(x) \leqslant t - \tau$
0 or 1 otherwise



- mutation pool size $p = O(\frac{log 1/\delta}{\delta})$
- sample size $S = O(\frac{\log 1/\delta}{\tau^2})$

Thanks!