# Attribute-Efficient Evolvability of Linear Functions

Varun Kanade

University of California, Berkeley

March 17, 2014

Joint work with Elaine Angelino (Harvard University)

# Outline

1. Summary : Previous Talks
2. Representation of Functions
3. Evolving Sparse Linear Functions
4. Conclusions and Future Work

# Evolution as computational learning

Leslie Valiant (2006)

- Genotype: string representation (*e.g.*, as encoded in DNA)
- Phenotype: function $X \to Y$
  - $(x_1, \ldots, x_n)$ — represents the environment
  - $y$ — desired output, *e.g.*, expression level of protein

# Evolution as computational learning

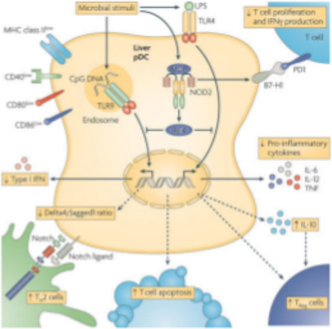
Leslie Valiant (2006)

- Genotype: string representation (*e.g.*, as encoded in DNA)
- Phenotype: function $X \rightarrow Y$
  - $(x_1, \ldots, x_n)$ — represents the environment
  - $y$ — desired output, *e.g.*, expression level of protein

Ideal function: best behaviour in each possible setting

For what classes of ideal functions is evolution feasible?

# Inside a Cell




From: Angus W. Thomson and Percy A. Kole, Nature Review Immunology 10, 753-766, 2010

- Snapshot of environment through sensors (*e.g.*, transcription factors)
- These factors affect gene production through interactions with DNA

# Last talk : From learning to evolution

- At a very high level (reductions of Feldman, P. Valiant)
  - representation encodes state of SQ learning algorithm, queries and their possible responses
  - selection "chooses" representation corresponding to correct query response

# Last talk : From learning to evolution

- At a very high level (reductions of Feldman, P. Valiant)
  - representation encodes state of SQ learning algorithm, queries and their possible responses
  - selection “chooses” representation corresponding to correct query response
- These mechanisms indeed fit in Valiant’s model
  - but the representations may be quite complex

# Outline

1 Summary : Previous Talks

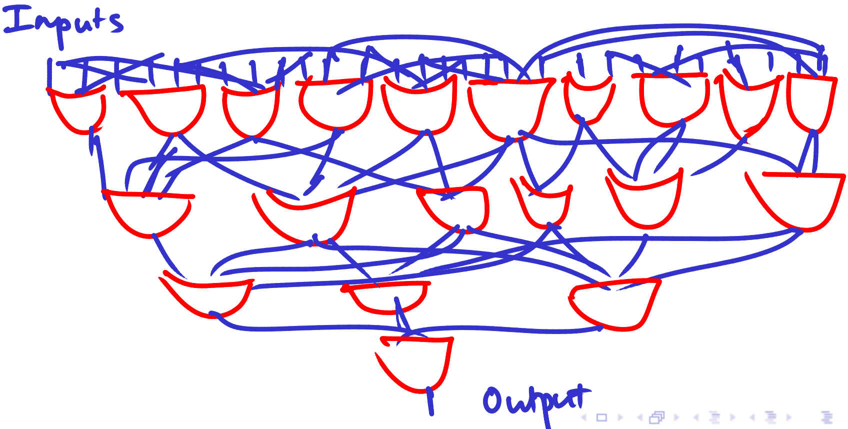
2 Representation of Functions

3 Evolving Sparse Linear Functions

4 Conclusions and Future Work

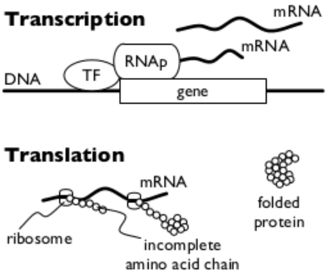
# Representation

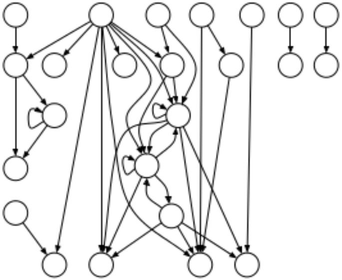
- Representation is a string describing a function (description of circuit)
  00011101011000110001100110000100011100001100000011110
- Valiant's model : Arbitrary circuit of polynomial size



# Gene Expression

- DNA is transcribed into mRNA, which is subsequently translated into protein
- Gene expression level is controlled by binding of RNA polymerase (RNAp)



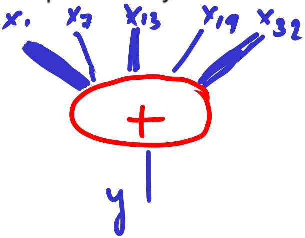



- Transcription factors (TFs) bind to the promoter region to activate/repress expression (by affecting binding of RNAp)

# Gene Expression Networks

- Transcription Networks in Prokaryotes:
  - The degree of networks is quite small, roughly 1 – 12 (short promoter region)
  - The depth of the network (cascade length) is also small (typically 1 – 4)
- Eukaryotic Regulation in more involved

- When viewed as circuits, small depth and fan-in
- Output depends only on a small number of input variables (juntas)



# Outline

1. Summary : Previous Talks
2. Representation of Functions
3. Evolving Sparse Linear Functions
4. Conclusions and Future Work

# Sparse Linear Functions

- Sparse linear functions
  - (Ideal) $f(x_1, \ldots, x_n) = 5x_1 + 7x_{17} - 3x_{45} + 100x_{100}$
  - Notion of performance (squared loss)

$$\mathrm{Perf}(r) = -\mathbb{E}_{\mathbf{x} \sim D}[(f(\mathbf{x}) - r(\mathbf{x}))^2]$$

# Sparse Linear Functions

- Sparse linear functions
  - (Ideal) $f(x_1, \ldots, x_n) = 5x_1 + 7x_{17} - 3x_{45} + 100x_{100}$
  - Notion of performance (squared loss)

$$\text{Perf}(r) = -\mathbb{E}_{\mathbf{x}\sim D}[(f(\mathbf{x}) - r(\mathbf{x}))^2]$$

- Evolutionary mechanism where each intermediate representation is a sparse linear function
  - sparse linear functions expressed as depth 1 weighted arithmetic circuit
  - representations may be less sparse than the ideal function

# Aside: Learning sparse linear functions

- Sparse linear regression (Machine Learning)
- Sparse signal recovery (Compressed Sensing)
- Sparsest solution to an underdetermined linear system of equations
- In general the problem is NP-hard
- However, if the distributions are somewhat “nice” the hardness results are broken

# Setting

Sparse Linear Functions:

$$\mathrm{Lin}^{k}_{l,u} = \{\mathbf{x} \mapsto \mathbf{w} \cdot \mathbf{x} \mid \mathrm{sparsity}(\mathbf{w}) \leq k, \forall i, w_i = 0 \text{ or } l \leq |w_i| \leq u\}$$

# Setting

Sparse Linear Functions:

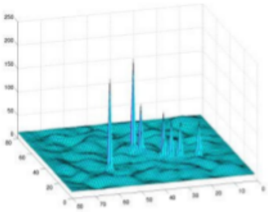
$$\mathrm{Lin}_{l,u}^{k} = \{\mathbf{x} \mapsto \mathbf{w} \cdot \mathbf{x} \mid \mathrm{sparsity}(\mathbf{w}) \leq k, \forall i, w_i = 0 \text{ or } l \leq |w_i| \leq u\}$$

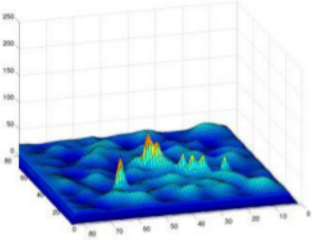
Smooth Distributions:

- Let $\tilde{D}$ be an arbitrary distribution over $\mathbb{R}^n$ (bounded support)
- Draw $\tilde{\mathbf{x}} \sim \tilde{D}$
- For each $i$, $x_i = \tilde{x}_i + \eta_i$, where $\eta_i \in [-\Delta, \Delta]$ uniformly at random
- $D$ is the smooth (noisy) distribution over $\mathbf{x}$
- Further, let $\mathbb{E}[x_i^2] \leq 1$ and support of $D$ is bounded

# Smooth Distributions

Inspired by work of Spielman and Teng





# Representations

- Representations also sparse linear functions

$$\text{Rep} = \{\mathbf{x} \mapsto \mathbf{w} \cdot \mathbf{x} \mid \mathrm{sparsity}(\mathbf{w}) \leq K, |w_i| \leq B\}$$

- Here $K, B$ depend on $k, u, l, \Delta$, (but not on $n$)

# Representations

- Representations also sparse linear functions

$$\text{Rep} = \{\mathbf{x} \mapsto \mathbf{w} \cdot \mathbf{x} \mid \text{sparsity}(\mathbf{w}) \leq K, |w_i| \leq B\}$$

- Here $K, B$ depend on $k, u, l, \Delta$, (but not on $n$)

Notation:

- Represent $r(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x}$ by vector $\mathbf{w}$
- $\text{NZ}(\mathbf{w}) = \{i \mid w_i \neq 0\}$
- Denote $\langle \mathbf{w}, \mathbf{w}' \rangle = \mathbb{E}_{x \sim D}[(\mathbf{w} \cdot \mathbf{x})(\mathbf{w}' \cdot \mathbf{x})]$
- Denote $\|\mathbf{w} - \mathbf{w}'\| = \mathbb{E}_{x \sim D}[(\mathbf{w} \cdot \mathbf{x} - \mathbf{w}' \cdot \mathbf{x})^2]$

# Properties of Smooth Distributions

- Let $D$ be a "smooth" (bounded) distribution over $\mathbb{R}^n$
- Let $\mathbf{w}$ be a vector representing function $\mathbf{x} \mapsto \mathbf{w} \cdot \mathbf{x}$
- Useful Properties:
  1. For each $i$, $w_i^2 \leq \frac{\langle \mathbf{w}, \mathbf{w} \rangle}{\Delta^2}$
  2. There exists an $i$, such that $w_i^2 \leq \frac{\langle \mathbf{w}, \mathbf{w} \rangle}{|\mathrm{NZ}(\mathbf{w})|\Delta^2}$
  3. $\mathbf{e}^i$ represents the function $x_i$, $\|\mathbf{e}^i\| = \Theta(\Delta)$

# Mutation Algorithm

- Sparse representation: $r(x_1, \ldots, x_n) = \sum_i w_i x_i$

# Mutation Algorithm

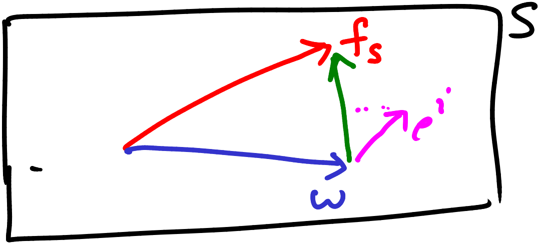
- Sparse representation: $r(x_1, \ldots, x_n) = \sum_i w_i x_i$
- Adjustments: (improve within current set of variables)
  1. Random rescaling:
  $$\mathbf{w} \leftarrow \alpha \mathbf{w} \text{ for some } \alpha \in [-1, 1]$$
  2. Reset a random coordinate
  $$\mathbf{w} \leftarrow \mathbf{w} - w_i \mathbf{e}^i + \beta_i \mathbf{e}^i \text{ for random } i \in \mathrm{NZ}(\mathbf{w}), \beta_i \in [-B, B]$$

# Mutation Algorithm

- Sparse representation: $r(x_1, \ldots, x_n) = \sum_i w_i x_i$
- Adjustments: (improve within current set of variables)
  1. Random rescaling:
  $$\mathbf{w} \leftarrow \alpha \mathbf{w} \text{ for some } \alpha \in [-1, 1]$$
  2. Reset a random coordinate
  $$\mathbf{w} \leftarrow \mathbf{w} - w_i \mathbf{e}^i + \beta_i \mathbf{e}^i \text{ for random } i \in \mathrm{NZ}(\mathbf{w}), \beta_i \in [-B, B]$$
- Jump improvements: (add new variables into consideration)
  1. Add a random coordinate
  $$\mathbf{w} \leftarrow \mathbf{w} + \beta_i \mathbf{e}^i \text{ for random } i \in [n] \setminus \mathrm{NZ}(\mathbf{w})$$
  2. Swap a random coordinate
  $$\mathbf{w} \leftarrow \mathbf{w} - w_i \mathbf{e}^i + \beta_j \mathbf{e}^j \text{ for random } i \in \mathrm{NZ}(\mathbf{w}), j \in [n] \setminus \mathrm{NZ}(\mathbf{w})$$

# Adjustment Mutations

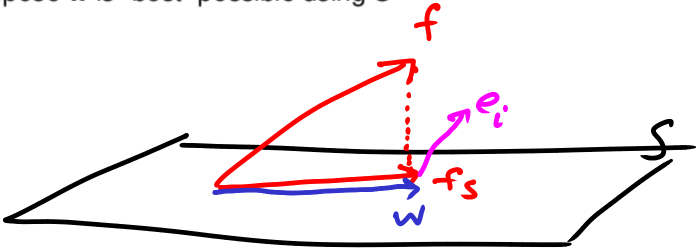
- Let $S = \text{NZ}(\mathbf{w})$ be the current set of non-zero variables in $\mathbf{w}$, $|S| \leq K$
- Either $\mathbf{w}$ is "best" possible using $S$
- Or adjustment to some co-ordinate is an improvement




- $\mathbf{f}_S$ best using variables in $S$, $\mathbf{r} = \mathbf{f}_S - \mathbf{w}$
- $\langle \mathbf{r}, \mathbf{r} \rangle = \sum_{i \in S} r_i \langle \mathbf{e}^i, \mathbf{r} \rangle$
- Beneficial Mutations Exist!

# Adding/Swapping a New Variable

- Let $S = \mathrm{NZ}(\mathbf{w})$ be the current set of non-zero variables in $\mathbf{w}$, $|S| \leq K$
- Suppose $\mathbf{w}$ is "best" possible using $S$




- $\mathbf{f}_S \approx \mathbf{w}$ best using variables in $S$, $\mathbf{r} = \mathbf{f} - \mathbf{w}$
- $\langle \mathbf{r}, \mathbf{r} \rangle = \sum_{i \in \mathrm{NZ}(\mathbf{f}) \setminus S} r_i \langle \mathbf{e}^i, \mathbf{r} \rangle$
- Some variable from $S$ can be discarded (has low influence)
- Beneficial Mutations Exist!

# Main result

**Theorem (Informal)**

*The class of sparse functions is evolvable under smooth distributions. Furthermore, it has the following strong attribute-efficient properties:*

- *the representations are all sparse linear functions*
- *the number of generations depends only on the sparsity of the ideal function and the accuracy $\epsilon$ (independent of n)*

*The population (number of mutations) at each generation is polynomial in n and* $1/\epsilon$.

# Outline


1. Summary : Previous Talks
2. Representation of Functions
3. Evolving Sparse Linear Functions
4. Conclusions and Future Work

# Conclusions and future work

Valiant's Framework and Computational Learning Theory provide a language to study several complexity measures for evolution

- This talk: sparsity inspired by transcription networks
- Other systems in biology: different constraints, similar analysis?
- Can richer classes of sparse functions be evolved?
  - sparse low-degree polynomials?
  - sparse linear functions with nonlinear filters?
    $f(x) = \mathrm{NL}(w \cdot x)$, where NL is a one-variable function
    *e.g.*, sigmoid, Hill, etc.
- Next Talk: What functions do gene expression levels represent?