Data Models and Deep Networks

Elchanan Mossel[1]

[1]MIT

Deep Learning Boot Camp Berkeley, May 2019

Why Deep Networks?

- Q: Why are deep networks successful?

Why Deep Networks?

- Q: Why are deep networks successful?
- A1: They have a lot of expressive power?
- A2: They generalize well?

Why Deep Networks?

- Q: Why are deep networks successful?
- A1: They have a lot of expressive power?
- But: why do we need to such power of expression? and how to find them?
- A2: They generalize well?
- But: Most of Theory is too general. Computational complexity unclear.

Why Deep Networks?

- Q: Why are deep networks successful?
- A1: They have a lot of expressive power?
- But: why do we need to such power of expression? and how to find them?
- A2: They generalize well?
- But: Most of Theory is too general. Computational complexity unclear.
- A3: It is about the Data.
- In particular, they work well and are needed on Data that is generated hierarchically.

Data Models and Deep Networks

Goal: Find data models that explain why deep nets work.

The Dream

- Goal: Find data models that explain why deep nets work.

The Dream

- Goal: Find data models that explain why deep nets work.
- $\implies$ understanding of why/when deep networks work.
- $\implies$ provable algorithms for inference.
- $\implies$ *robust* provable algorithms for inference.
- $\implies$ Proof that depth is needed.

Criteria

- 3 important criteria from a theory perspective:

Criteria

- 3 important criteria from a theory perspective:
- 1. Realism: Reasonable data models.

Criteria

- 3 important criteria from a theory perspective:
- 1. Realism: Reasonable data models.
- 2. Reconstruction: Provable efficient algorithms for inference.

Criteria

- 3 important criteria from a theory perspective:
- 1. Realism: Reasonable data models.
- 2. Reconstruction: Provable efficient algorithms for inference.
- 3. Depth: Proof that depth is needed.

Criteria

- 3 important criteria from a theory perspective:
- 1. Realism: Reasonable data models.
- 2. Reconstruction: Provable efficient algorithms for inference.
- 3. Depth: Proof that depth is needed.
- Next we will explore some models suggested along this axis.

Candidate 1: The Pure Theorist Model

- TCS: Data: (x_i, y_i), where x_i are i.i.d. $\sim U(\{-1,1\}^n)$ and
- $y_i = f(x_i)$ where $f = poly(n)$ size depth d circuit.
- Circuit has (unbounded fan) AND/OR/NOT gates.

Candidate 1: The Pure Theorist Model

- TCS: Data: (x_i, y_i), where x_i are i.i.d. $\sim U(\{-1,1\}^n)$ and
- $y_i = f(x_i)$ where $f = poly(n)$ size depth d circuit.
- Circuit has (unbounded fan) AND/OR/NOT gates.
- Thm(Hastad, Rossman, Servedio, Tan):
- An explicit $O(n)$ size depth d circuit labeling the data s.t.:
- Any circuits g depth $d-1$ with $P[g(x) \neq f(x)] \leq 0.5 - \varepsilon$ must be of size $\exp(n^{\Omega(1/d)})$.

Candidate 1: The Pure Theorist Model

- TCS: Data: (x_i, y_i), where x_i are i.i.d. $\sim U(\{-1,1\}^n)$ and
- $y_i = f(x_i)$ where $f = poly(n)$ size depth d circuit.
- Circuit has (unbounded fan) AND/OR/NOT gates.
- Thm(Hastad, Rossman, Servedio, Tan):
- An explicit $O(n)$ size depth d circuit labeling the data s.t.:
- Any circuits g depth $d-1$ with $P[g(x) \neq f(x)] \leq 0.5 - \varepsilon$ must be of size $\exp(n^{\Omega(1/d)})$.
- Score?

Candidate 1: The Pure Theorist Model

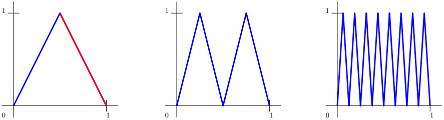
- TCS: Data: (x_i, y_i), where x_i are i.i.d. $\sim U(\{-1,1\}^n)$ and
- $y_i = f(x_i)$ where $f = poly(n)$ size depth d circuit.
- Circuit has (unbounded fan) AND/OR/NOT gates.
- Thm(Hastad, Rossman, Servedio, Tan):
- An explicit $O(n)$ size depth d circuit labeling the data s.t.:
- Any circuits g depth $d-1$ with $P[g(x) \neq f(x)] \leq 0.5 - \varepsilon$ must be of size $\exp(n^{\Omega(1/d)})$.
- Score?
- Score: Depth: 10, Reconstruction: 0, Realsim: 0.

Candidate 1': A DL Theorist Perspective

Slide by Telgarsky:

Consider the **tent map**

$$\Delta(x) := \sigma_{\mathrm{r}}(2x) - \sigma_{\mathrm{r}}(4x-2) = \begin{cases} 2x & x \in [0, 1/2), \\ 2(1-x) & x \in [1/2, 1]. \end{cases}$$



$\Delta.$ $\quad\quad$ $\Delta^2 = \Delta \circ \Delta.$ $\quad\quad$ $\Delta^k.$

What is the effect of composition?

$$f(\Delta(x)) = \begin{cases} x \in [0, 1/2) & \Longrightarrow \quad f(2x) = f \text{ squeezed into } [0, 1/2], \\ x \in [1/2, 1] & \Longrightarrow \quad f\big(2(1-x)\big) = f \text{ reversed, squeezed.} \end{cases}$$

Δ^k uses $\mathcal{O}(k)$ layers & nodes, but has $\mathcal{O}(2^k)$ bumps.

Candidate 1': A DL Theorist Perspective

- Telgarsky: $x_i \sim U[0,1]$ and $y_i = f(x_i)$.

Candidate 1': A DL Theorist Perspective

- Telgarsky: $x_i \sim U[0,1]$ and $y_i = f(x_i)$.
- Thm(Telgarsky):
- Explicit $O(d^2)$ size/depth RELU f net labelling the data s.t.
- Any RELU net g of depth d has $E[|f(x) - g(x)] \leq 0.01$ must be of size $\exp(\Omega(d))$.

Candidate 1': A DL Theorist Perspective

- Telgarsky: $x_i \sim U[0,1]$ and $y_i = f(x_i)$.
- Thm(Telgarsky):
- Explicit $O(d^2)$ size/depth RELU f net labelling the data s.t.
- Any RELU net g of depth d has $E[|f(x) - g(x)] \leq 0.01$ must be of size $\exp(\Omega(d))$.
- Score?

Candidate 1': A DL Theorist Perspective

- Telgarsky: $x_i \sim U[0,1]$ and $y_i = f(x_i)$.
- Thm(Telgarsky):
- Explicit $O(d^2)$ size/depth RELU f net labelling the data s.t.
- Any RELU net g of depth d has $E[|f(x) - g(x)] \leq 0.01$ must be of size $\exp(\Omega(d))$.
- Score?
- Score: Depth: 9, Reconstruction: 0, Realsim: 2.

Candidate 1': A DL Theorist Perspective

- Telgarsky: $x_i \sim U[0,1]$ and $y_i = f(x_i)$.
- Thm(Telgarsky):
- Explicit $O(d^2)$ size/depth RELU f net labelling the data s.t.
- Any RELU net g of depth d has $E[|f(x) - g(x)] \leq 0.01$ must be of size $\exp(\Omega(d))$.
- Score?
- Score: Depth: 9, Reconstruction: 0, Realsim: 2.
- Proof is elegant :)

Candidate 2: Hacker models

- Data is also generated by a network:
- Ex 1: Reversible models: data: $\Downarrow$, inference: $\Uparrow$.
- Ex 2: GANS (Goodfellow), Variational Auto encoders, ...

Candidate 2: Hacker models

- Data is also generated by a network:
- Ex 1: Reversible models: data: $\Downarrow$, inference: $\Uparrow$.
- Ex 2: GANS (Goodfellow), Variational Auto encoders, ...
- Score?

Candidate 2: Hacker models

- Data is also generated by a network:
- Ex 1: Reversible models: data: $\Downarrow$, inference: $\Uparrow$.
- Ex 2: GANS (Goodfellow), Variational Auto encoders, ...
- Score?
- Score: Realsim: 9, Reconstruction: 2, Depth: 2.

Candidate 2': Theory Hacker model

- Arora et al: Random, Sparse generative models.

Candidate 2': Theory Hacker model

- Arora et al: Random, Sparse generative models.
- Thm: Each pair of layers is a (noisy) auto encoder.
- Thm: Efficient algorithm for learning the network.

Candidate 2': Theory Hacker model

- Arora et al: Random, Sparse generative models.
- Thm: Each pair of layers is a (noisy) auto encoder.
- Thm: Efficient algorithm for learning the network.
- Score?

Candidate 2': Theory Hacker model

- Arora et al: Random, Sparse generative models.
- Thm: Each pair of layers is a (noisy) auto encoder.
- Thm: Efficient algorithm for learning the network.
- Score?
- Score: Realsim: 5, Reconstruction: 9, Depth: 4.

Some intuition

- Sparsity + Randomness $\implies$ unique neighbor property $\implies$
- if a node is 1 at level 2 most of its neighbors at level 1 have it as the only neighbor that is on.

Some intuition

- Sparsity + Randomness $\implies$ unique neighbor property $\implies$
- if a node is 1 at level 2 most of its neighbors at level 1 have it as the only neighbor that is on.
- $\implies$ auto-encoding property.
- $\implies$ sisters/brothers tend to fire together.

Some intuition

- Sparsity + Randomness $\implies$ unique neighbor property $\implies$
- if a node is 1 at level 2 most of its neighbors at level 1 have it as the only neighbor that is on.
- $\implies$ auto-encoding property.
- $\implies$ sisters/brothers tend to fire together.
- Hebb: "Things that fire together wire together"
- Also: a key property in recovery tree graphical models (Neighbor Joining ...)

Candidate 3: Scattering Transform

- Mallat, Bruna+Mallat: Data generative model S that is:

Candidate 3: Scattering Transform

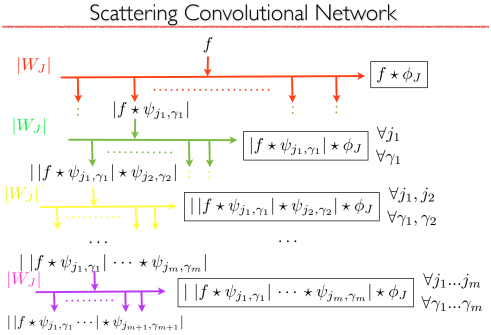
- Mallat, Bruna+Mallat: Data generative model S that is:
- Continuous with respect to natural geometric deformations at different scales.

Candidate 3: Scattering Transform

- Mallat, Bruna+Mallat: Data generative model S that is:
- Continuous with respect to natural geometric deformations at different scales.
- Generative process: Energy moves from high frequencies to low frequencies.

Candidate 3: Scattering Transform

Slide from Joan Bruna:



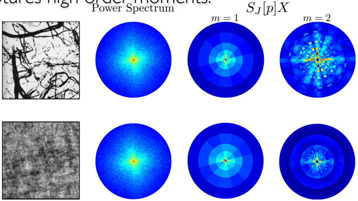
Cascade of contractive operators.

Candidate 3: Scattering Transform

Slide from Joan Bruna:

Properties of Scattering Moments

- Captures high order moments: [Bruna, Mallat, '11,'12]



- Cascading non-linearities is ***necessary*** to reveal higher-order moments.

Candidate 3: Scattering Transform

- Mallat, Bruna+Mallat: Data generative model S that is:
- Continuous with respect to natural geometric deformations at different scales.
- Generative process: Energy moves from high frequencies to low frequencies.

Candidate 3: Scattering Transform

- Mallat, Bruna+Mallat: Data generative model S that is:
- Continuous with respect to natural geometric deformations at different scales.
- Generative process: Energy moves from high frequencies to low frequencies.
- Inference process: High frequency information is recovered via a recursive non-linear procedure.

Candidate 3: Scattering Transform

- Mallat, Bruna+Mallat: Data generative model S that is:
- Continuous with respect to natural geometric deformations at different scales.
- Generative process: Energy moves from high frequencies to low frequencies.
- Inference process: High frequency information is recovered via a recursive non-linear procedure.
- Depth: Bruna: "Cascading non-linearities is necessary to reveal higher- order moments"
- But this is not completely formal.

Candidate 3: Scattering Transform

- Mallat, Bruna+Mallat: Data generative model S that is:
- Continuous with respect to natural geometric deformations at different scales.
- Generative process: Energy moves from high frequencies to low frequencies.
- Inference process: High frequency information is recovered via a recursive non-linear procedure.
- Depth: Bruna: "Cascading non-linearities is necessary to reveal higher- order moments"
- But this is not completely formal.
- Score?

Candidate 3: Scattering Transform

- Mallat, Bruna+Mallat: Data generative model S that is:
- Continuous with respect to natural geometric deformations at different scales.
- Generative process: Energy moves from high frequencies to low frequencies.
- Inference process: High frequency information is recovered via a recursive non-linear procedure.
- Depth: Bruna: "Cascading non-linearities is necessary to reveal higher- order moments"
- But this is not completely formal.
- Score?
- Score: Realsim: 8, Reconstruction: 5 (see e.g. Cohen and Welling), Depth: 5.

The Question Remains

- Q: Is there

The Question Remains

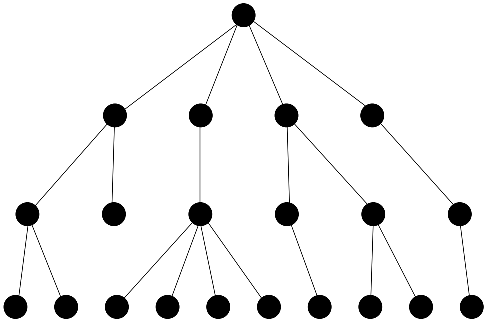
- Q: Is there
- A natural data generative process with
- Provable algorithms for learning classifier?
- and where classifier provably requires depth?

The Question Remains

- Q: Is there
- A natural data generative process with
- Provable algorithms for learning classifier?
- and where classifier provably requires depth?
- It would be nice if classifier runs in linear time.

Information Flow on Trees

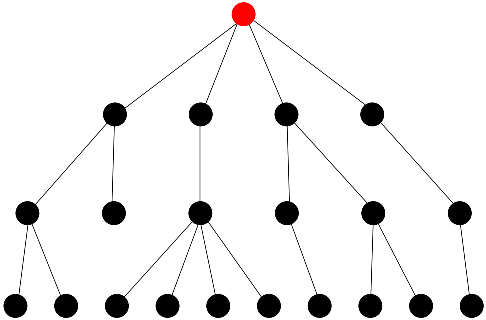
Consider the following process on a tree.



Information Flow on Trees

Consider the following process on a tree.

Color the root randomly.

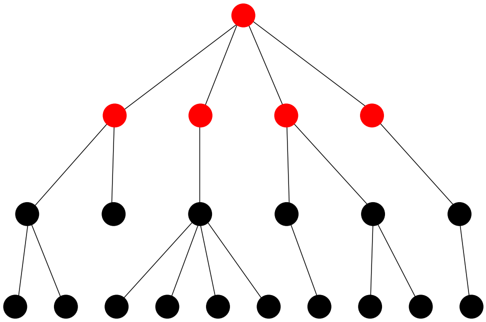


Information Flow on Trees

Consider the following process on a tree.

Color the root randomly.

Repeat: Copy color of parent with probability θ. Otherwise, chose color $\sim U[q]$.

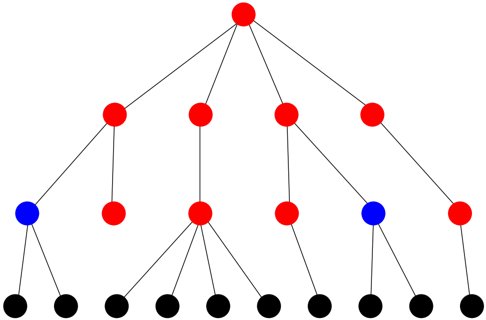


Information Flow on Trees

Consider the following process on a tree.

Color the root randomly.

Repeat: Copy color of parent with probability θ. Otherwise, chose color $\sim U[q]$.



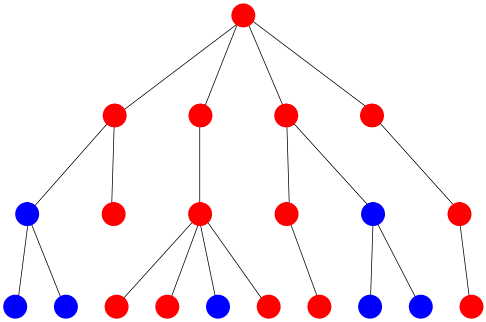
Information Flow on Trees

Consider the following process on a tree.

Color the root randomly.

Repeat: Copy color of parent with probability θ. Otherwise, chose color $\sim U[q]$.

In this talk, we will only consider full binary trees.



Information Flow on Trees

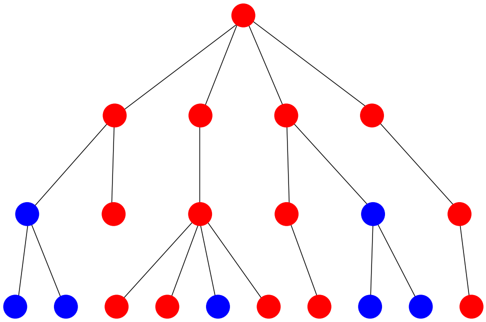
Consider the following process on a tree.

Color the root randomly.

Repeat: Copy color of parent with probability θ. Otherwise, chose color $\sim U[q]$.

In this talk, we will only consider full binary trees.

More generally, we can consider any Markov chain along the edges and $\theta =$ 2nd eigenvalue of transition matrix



Is this process natural?

- Bad model of speech/images.

Is this process natural?

- Bad model of speech/images.
- But: Nice abstraction of multi-layer generative processes.

Is this process natural?

- Bad model of speech/images.
- But: Nice abstraction of multi-layer generative processes.
- In fact, standard model of evolutionary dynamics of species (since 1970s: Cavender-Farris-Neyman model).

Is this process natural?

- Bad model of speech/images.
- But: Nice abstraction of multi-layer generative processes.
- In fact, standard model of evolutionary dynamics of species (since 1970s: Cavender-Farris-Neyman model).
- A standard multi-layered model in statistical physics (since the 1920s: Ising Model on the Bethe lattice).

Is this process natural?

- Bad model of speech/images.
- But: Nice abstraction of multi-layer generative processes.
- In fact, standard model of evolutionary dynamics of species (since 1970s: Cavender-Farris-Neyman model).
- A standard multi-layered model in statistical physics (since the 1920s: Ising Model on the Bethe lattice).
- Realsim?

Is this process natural?

- Bad model of speech/images.
- But: Nice abstraction of multi-layer generative processes.
- In fact, standard model of evolutionary dynamics of species (since 1970s: Cavender-Farris-Neyman model).
- A standard multi-layered model in statistical physics (since the 1920s: Ising Model on the Bethe lattice).
- Realsim?
- Overall: Realism: 6.

What is the best classifier

- Compute Bayes Posterior by running Belief Propagation.

What is the best classifier

- Compute Bayes Posterior by running Belief Propagation.
- Runs in linear time.

Provable Algorithms for learning classifier

- Learning the classifier is the same as learning Phylogenies. Much is known in very general setups.

Provable Algorithms for learning classifier

- Learning the classifier is the same as learning Phylogenies. Much is known in very general setups.
- ESSW 90s: Polynomial time algorithm for learning graphical model.
- M-04, M-Steel-05, DMR-10: Phase transitions for sampling complexity from logarithmic to polynomial.
- M-Roch-05: PAC learning.

Provable Algorithms for learning classifier

- Learning the classifier is the same as learning Phylogenies. Much is known in very general setups.
- ESSW 90s: Polynomial time algorithm for learning graphical model.
- M-04, M-Steel-05, DMR-10: Phase transitions for sampling complexity from logarithmic to polynomial.
- M-Roch-05: PAC learning.
- Reconstruction Score?

Provable Algorithms for learning classifier

- Learning the classifier is the same as learning Phylogenies. Much is known in very general setups.
- ESSW 90s: Polynomial time algorithm for learning graphical model.
- M-04, M-Steel-05, DMR-10: Phase transitions for sampling complexity from logarithmic to polynomial.
- M-Roch-05: PAC learning.
- Reconstruction Score?
- Reconstruction Score: 9.

Depth Lower bounds

- Note that the generative model has depth $O(\log n)$, so cannot expect better than $\log n$ lower bounds.

Depth Lower bounds

- Note that the generative model has depth $O(\log n)$, so cannot expect better than $\log n$ lower bounds.
- Maybe the right scaling?

Depth Lower bounds

- Note that the generative model has depth $O(\log n)$, so cannot expect better than $\log n$ lower bounds.
- Maybe the right scaling?
- VGGNet (from 2014) has 16 layers with $> 100M$ parameters.

Depth Lower bounds

- Note that the generative model has depth $O(\log n)$, so cannot expect better than $\log n$ lower bounds.
- Maybe the right scaling?
- VGGNet (from 2014) has 16 layers with $> 100M$ parameters.
- Maybe not: ResNet (from 2015) beat it with 152 layers but only $2M$ parameters.

Depth Lower bounds

- Note that the generative model has depth $O(\log n)$, so cannot expect better than $\log n$ lower bounds.
- Maybe the right scaling?
- VGGNet (from 2014) has 16 layers with $> 100M$ parameters.
- Maybe not: ResNet (from 2015) beat it with 152 layers but only $2M$ parameters.
- Next we will discuss some recent depth lower bound for this model (Moitra-M-Sandon).

$\mathbf{AC}^0$

- $\mathbf{AC}^0$:= class of bounded depth circuits with AND/OR (unbounded fan) and NOT gates.

$\mathbf{AC}^0$

- $\mathbf{AC}^0$:= class of bounded depth circuits with AND/OR (unbounded fan) and NOT gates.
- Thm: Moitra-M-Sandon-19:
- $\mathbf{AC}^0$ cannot classify better than random.

$\mathbf{AC}^0$

- $\mathbf{AC}^0$:= class of bounded depth circuits with AND/OR (unbounded fan) and NOT gates.
- Thm: Moitra-M-Sandon-19:
- $\mathbf{AC}^0$ cannot classify better than random.
- Is this trivial?

$\mathbf{AC}^0$

- $\mathbf{AC}^0 :=$ class of bounded depth circuits with AND/OR (unbounded fan) and NOT gates.
- Thm: Moitra-M-Sandon-19:
- $\mathbf{AC}^0$ cannot classify better than random.
- Is this trivial?
- Maybe not: Known that BP classifies better than random, when $2\theta^2 > 1$.
- Also: Thm MMS-19: $\mathbf{AC}^0$ generates leaf distributions.

$\mathbf{TC}^0$

- $\mathbf{TC}^0 :=$ like $\mathbf{AC}^0$ but with Majority gates.
- "Bounded depth deep nets".

$\mathbf{TC}^0$

- $\mathbf{TC}^0$:= like $\mathbf{AC}^0$ but with Majority gates.
- "Bounded depth deep nets".
- Thm (MMS-19): When $q = 2$ and $\theta < 1$ is large enough, $\mathbf{TC}^0$ classifies as well as BP.
- Conj: This is true for all θ when $q = 2$.

$\mathbf{TC}^0$

- $\mathbf{TC}^0$:= like $\mathbf{AC}^0$ but with Majority gates.
- "Bounded depth deep nets".
- Thm (MMS-19): When $q = 2$ and $\theta < 1$ is large enough, $\mathbf{TC}^0$ classifies as well as BP.
- Conj: This is true for all θ when $q = 2$.
- So maybe we can classify optimally in $\mathbf{TC}^0$?
- Maybe bounded depth nets suffice?

$\mathbf{NC}^1$

- $\mathbf{NC}^1 :=$ class of $O(\log n)$ depth circuits with AND/OR (fan 2) and NOT gates.

$\mathbf{NC}^1$

- $\mathbf{NC}^1 :=$ class of $O(\log n)$ depth circuits with AND/OR (fan 2) and NOT gates.
- Known that $\mathbf{TC}^0 \subset \mathbf{NC}^1$. Open if they are the same.

$\mathbf{NC}^1$

- $\mathbf{NC}^1 :=$ class of $O(\log n)$ depth circuits with AND/OR (fan 2) and NOT gates.
- Known that $\mathbf{TC}^0 \subset \mathbf{NC}^1$. Open if they are the same.
- Thm (MMS-19): One can classify as well as BP in $\mathbf{NC}^1$.

$\mathbf{NC}^1$

- $\mathbf{NC}^1$:= class of $O(\log n)$ depth circuits with AND/OR (fan 2) and NOT gates.
- Known that $\mathbf{TC}^0 \subset \mathbf{NC}^1$. Open if they are the same.
- Thm (MMS-19): One can classify as well as BP in $\mathbf{NC}^1$.
- Thm (MMS-19): There is a broadcast process for which classifying better than random is $\mathbf{NC}^1$-complete.

$\mathbf{NC}^1$

- $\mathbf{NC}^1 :=$ class of $O(\log n)$ depth circuits with AND/OR (fan 2) and NOT gates.
- Known that $\mathbf{TC}^0 \subset \mathbf{NC}^1$. Open if they are the same.
- Thm (MMS-19): One can classify as well as BP in $\mathbf{NC}^1$.
- Thm (MMS-19): There is a broadcast process for which classifying better than random is $\mathbf{NC}^1$-complete.
- So, unless $\mathbf{TC}^0 = \mathbf{NC}^1$, $\log n$ depth is needed.

$\mathbf{NC}^1$

- $\mathbf{NC}^1 :=$ class of $O(\log n)$ depth circuits with AND/OR (fan 2) and NOT gates.
- Known that $\mathbf{TC}^0 \subset \mathbf{NC}^1$. Open if they are the same.
- Thm (MMS-19): One can classify as well as BP in $\mathbf{NC}^1$.
- Thm (MMS-19): There is a broadcast process for which classifying better than random is $\mathbf{NC}^1$-complete.
- So, unless $\mathbf{TC}^0 = \mathbf{NC}^1$, $\log n$ depth is needed.
- Depth score?

$\mathbf{NC}^1$

- $\mathbf{NC}^1$:= class of $O(\log n)$ depth circuits with AND/OR (fan 2) and NOT gates.
- Known that $\mathbf{TC}^0 \subset \mathbf{NC}^1$. Open if they are the same.
- Thm (MMS-19): One can classify as well as BP in $\mathbf{NC}^1$.
- Thm (MMS-19): There is a broadcast process for which classifying better than random is $\mathbf{NC}^1$-complete.
- So, unless $\mathbf{TC}^0 = \mathbf{NC}^1$, $\log n$ depth is needed.
- Depth score?
- Depth score: 7.

The KS bound and Circuit Complexity

- The threshold $2\theta^2 = 1$ is called the Kesten-Stigum threshold.

The KS bound and Circuit Complexity

- The threshold $2\theta^2 = 1$ is called the Kesten-Stigum threshold.
- Above this threshold it is known that one neuron can classify the root better than random (Kesten-Stigum-66).
- Below this threshold, one neuron cannot (M-Peres-04).
- Below this threshold, with enough i.i.d. noise on the leaves, BP becomes trivial (Janson-M-05).

The KS bound and Circuit Complexity

- The threshold $2\theta^2 = 1$ is called the Kesten-Stigum threshold.
- Above this threshold it is known that one neuron can classify the root better than random (Kesten-Stigum-66).
- Below this threshold, one neuron cannot (M-Peres-04).
- Below this threshold, with enough i.i.d. noise on the leaves, BP becomes trivial (Janson-M-05).
- Related to "Replica Symmetry Breaking" in statistical physics models (Mezard-Montanari-06).

The KS bound and Circuit Complexity

- The threshold $2\theta^2 = 1$ is called the Kesten-Stigum threshold.
- Above this threshold it is known that one neuron can classify the root better than random (Kesten-Stigum-66).
- Below this threshold, one neuron cannot (M-Peres-04).
- Below this threshold, with enough i.i.d. noise on the leaves, BP becomes trivial (Janson-M-05).
- Related to "Replica Symmetry Breaking" in statistical physics models (Mezard-Montanari-06).
- Conjecture (Moitra-M-Sandon): For any broadcast process, below the KS bound and where BP classifies better than random, classification is $\mathbf{NC}^1$-complete.

Some intuition for $\mathbf{AC}^0$ hardness

- Standard technique in lower bounds: apply random restrictions and show that
  - Circuit becomes constant.
  - Your function is not.

Some intuition for $\mathbf{AC}^0$ hardness

- Standard technique in lower bounds: apply random restrictions and show that
  - Circuit becomes constant.
  - Your function is not.
- The tree broadcast process provides natural recursive random restrictions:
  - Each child gets value 0 or 1 with probability $(1-\theta)/2$.
  - All other children are assigned the same value as the root.

Some intuition for $\mathbf{AC}^0$ hardness

- Standard technique in lower bounds: apply random restrictions and show that
  - Circuit becomes constant.
  - Your function is not.
- The tree broadcast process provides natural recursive random restrictions:
  - Each child gets value 0 or 1 with probability $(1-\theta)/2$.
  - All other children are assigned the same value as the root.
- To generate: Go over all noise patterns that result in a certain value.

Some intuition for $\mathbf{TC}^0$ results

- Circuit construction:
  - Perform majority on big sub-trees.
  - Run constant level BP on majorities.

Some intuition for $\mathbf{TC}^0$ results

- Circuit construction:
  - Perform majority on big sub-trees.
  - Run constant level BP on majorities.
- Technical ingredient (M-Neeman-Sly-14): BP with noise classifies as well as BP without noise if θ close enough to 1 and $q = 2$.

Some intuition for **NC**^{1} hardness

- Start from the word problem over A_5 which is known to be **NC**^{1}-complete.

Some intuition for $\mathbf{NC}^1$ hardness

- Start from the word problem over A_5 which is known to be $\mathbf{NC}^1$-complete.
- Show that an average version of the problem is also $\mathbf{NC}^1$-complete.

Some intuition for $\mathbf{NC}^1$ hardness

- Start from the word problem over A_5 which is known to be $\mathbf{NC}^1$-complete.
- Show that an average version of the problem is also $\mathbf{NC}^1$-complete.
- Show that BP for an appropriate broadcast process solves this problem.

Some intuition for $\mathbf{NC}^1$ hardness

- Start from the word problem over A_5 which is known to be $\mathbf{NC}^1$-complete.
- Show that an average version of the problem is also $\mathbf{NC}^1$-complete.
- Show that BP for an appropriate broadcast process solves this problem.
- Interestingly, broadcast process has second eigenvalue 0.

The KS bound and Circuit Complexity

- The threshold $2\theta^2 = 1$ is called the Kesten-Stigum threshold.
- Above this threshold it is known that one neuron can classify the root better than random (Kesten-Stigum).
- Below this threshold, one neuron cannot (M-Peres).
- Below this threshold, with enough i.i.d. noise on the leaves, BP becomes trivial (Janson-M).
- Related to "Replica Symmetry Breaking" in statistical physics model (Mezard-Montanari-06).

The KS bound and Circuit Complexity

- The threshold $2\theta^2 = 1$ is called the Kesten-Stigum threshold.
- Above this threshold it is known that one neuron can classify the root better than random (Kesten-Stigum).
- Below this threshold, one neuron cannot (M-Peres).
- Below this threshold, with enough i.i.d. noise on the leaves, BP becomes trivial (Janson-M).
- Related to "Replica Symmetry Breaking" in statistical physics model (Mezard-Montanari-06).
- Conjecture (Moitra-M-Sandon): For any broadcast process, below the KS bound and where BP classifies better than random, classification is $\mathbf{NC}^1$-complete.

Another model where the KS bound plays a role

Next we will discuss a related semi-supervised structure learning where the KS bound plays a role.

Phylogenetic Reconstruction

- In Phylogenetic Reconstruction, want to reconstruct the a *Tree* T.

Phylogenetic Reconstruction

- In Phylogenetic Reconstruction, want to reconstruct the a *Tree* T.
- When
  - T is a binary ($d = 2$) tree and
  - $Data =$ sequences of colors $\in [q]$ at leaves.

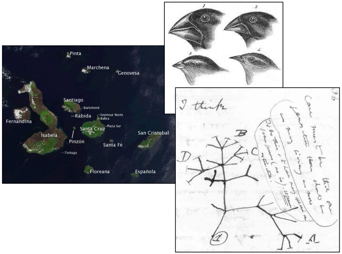
Phylogenetic Reconstruction

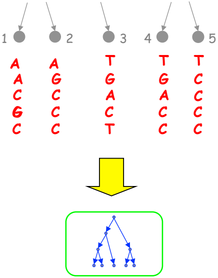
- In Phylogenetic Reconstruction, want to reconstruct the a *Tree* T.
- When
  - T is a binary ($d = 2$) tree and
  - *Data* = sequences of colors $\in [q]$ at leaves.
- Sequences of colors are generated from the broadcast process above.

Phylogenetic Reconstruction

- In Phylogenetic Reconstruction, want to reconstruct the a *Tree* T.
- When
  - T is a binary ($d = 2$) tree and
  - *Data* $=$ sequences of colors $\in [q]$ at leaves.
- Sequences of colors are generated from the broadcast process above.
- E.G. $q = 4$ and colors are A, C, G and T.

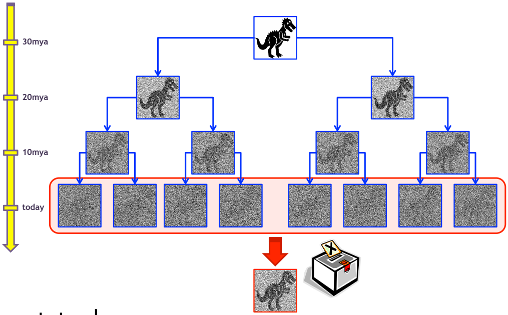
The Phylogenetic Inference Problem





Broadcasting on trees and Phylogenetic trees

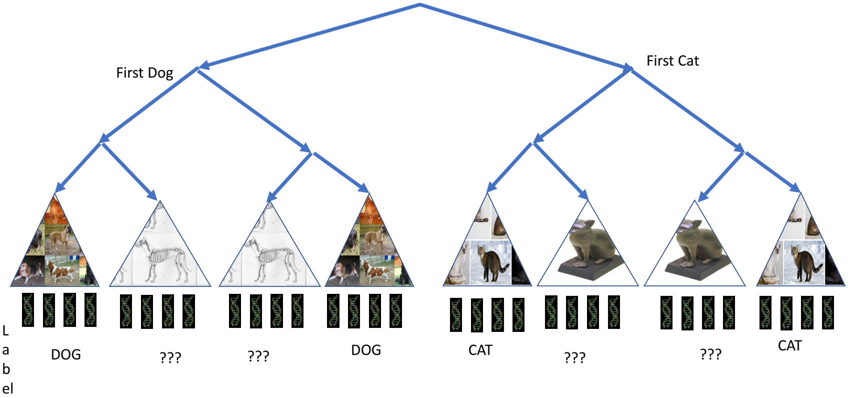
Picture courtesy of Costis Daskalakis



Three different tasks

- Reconstruction: Given a known tree, reconstruct ancestral sequence from sequences at the leaves.
- Phylogeny Recovery: Given sequences reconstruct the tree.
- Semi-supervised learning:

A semi supervised setting



Shallow Algorithms

Theorem (M-04 ... ; M-16)

Suppose that $2\theta^2 > 1$ then for all q there is an algorithm that labels all labelled data correctly. Moreover, this algorithm is shallow.

Theorem (M-16)

Suppose that $2\theta^2 < 1$ then it is information theoretically impossible to classify better than random.

- A Shallow algorithm cannot use the correlation between different features in the labelled data.

Shallow Algorithms

Theorem (M-04 ... ; M-16)

Suppose that $2\theta^2 > 1$ then for all q there is an algorithm that labels all labelled data correctly. Moreover, this algorithm is shallow*.*

Theorem (M-16)

Suppose that $2\theta^2 < 1$ then it is information theoretically impossible to classify better than random.

- A Shallow algorithm cannot use the correlation between different features in the labelled data.
- Can use all the unlabelled data.

Deep Algorithms

Theorem (M-16)

Suppose that $2\theta^2 < 1$ *then it is information theoretically impossible for any shallow algorithm to label* 0.6 *of the unlabelled data correctly.*

Theorem (M-16)

Suppose that $2\theta > 1$ *and* q *is large enough, then then it is possible to label all the unlabelled data correctly.*

Deep Algorithms

Theorem (M-16)

Suppose that $2\theta^2 < 1$ *then it is information theoretically impossible for any shallow algorithm to label* 0.6 *of the unlabelled data correctly.*

Theorem (M-16)

Suppose that $2\theta > 1$ *and* q *is large enough, then then it is possible to label all the unlabelled data correctly.*

- Separation between deep and shallow learning.

What is a shallow algorithm?

- A shallow algorithm is an algorithm that cannot use interaction between the features of the labelled data. More formally:
- Let A denote the unlabelled data and B denote the labelled data.
- The input to the shallow algorithm is:

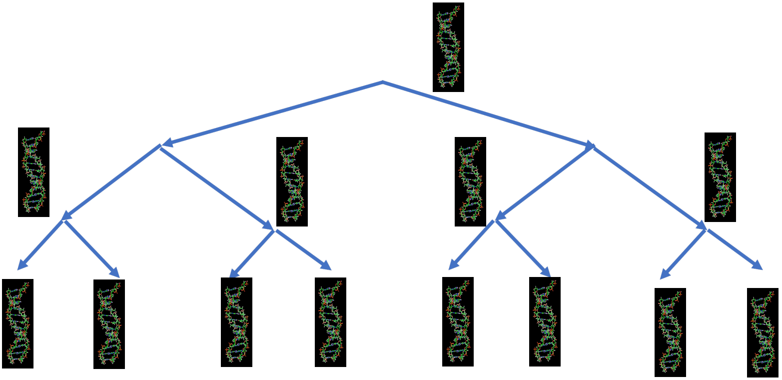
$$\Big(\sigma^h(u) : u \in A\Big),$$

$$\Big(n_\ell(j,a) : a, 1 \leq j \leq k\Big), \quad n_\ell(j,a) := \Big|\{v : v \in B, L(v) = \ell, \sigma_j^v = a\}$$

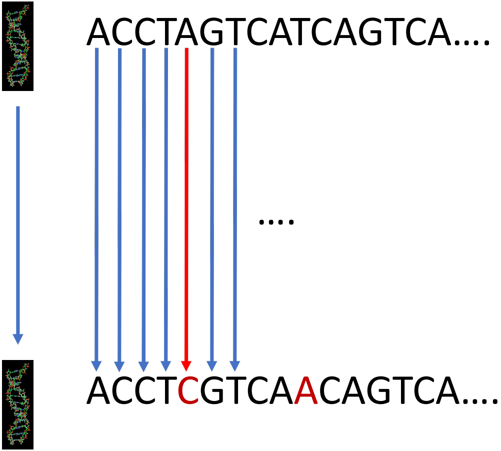
More complex models?

Do the same results hold for more complex models?

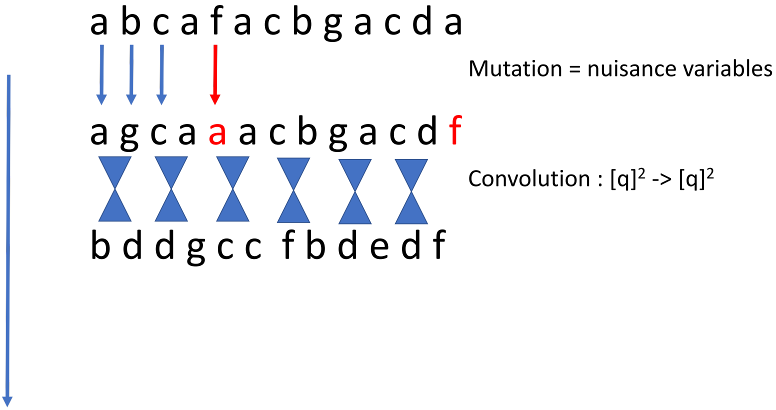
The phylogenetic Model Zoom Out



The phylogenetic Model Zoom In



Adding Interaction Between Features



Deep Algorithms

The following two theorems hold also when adding interaction between features.

Theorem (M-16)

Suppose that $2\theta^2 < 1$ *then it is information theoretically impossible for any shallow algorithm to label* 0.6 *of the unlabelled data correctly.*

Theorem (M-16)

Suppose that $2\theta > 1$ *and* q *is large enough, then then it is possible to label all the unlabelled data correctly.*

- Separation between deep and shallow learning!

Deep Algorithms

The following two theorems hold also when adding interaction between features.

Theorem (M-16)

Suppose that $2\theta^2 < 1$ *then it is information theoretically impossible for any shallow algorithm to label* 0.6 *of the unlabelled data correctly.*

Theorem (M-16)

Suppose that $2\theta > 1$ *and* q *is large enough, then then it is possible to label all the unlabelled data correctly.*

- Separation between deep and shallow learning!
- Conjecture: Separation is typically much stronger.

Natural Challenges

- More realistic models and testing on data?
- E.G: Malach-Shalev Schwartz (18) - image models with provable reconstruction algorithms.
- But no depth lower bounds.

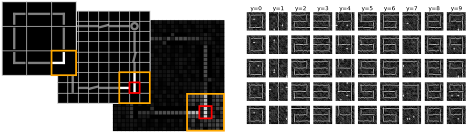


Figure 2: Left: Image generation process example. Right: Synthetic examples generated.

the lower-level image. If we succeed in doing so multiple times, we can infer the topmost semantic image in the hierarchy. Assuming the high-level distribution $\mathcal{G}_0$ is simple enough (for example, a linearly separable distribution with respect to some embedding of the classes), we could then use a simple classification algorithm on the high-level image to infer its label.

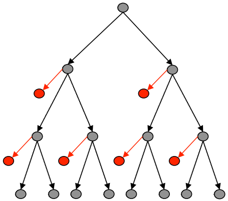
Unfortunately, we cannot learn these semantic classes directly as we are not given access to the latent semantic images, but only to the lowest-level image generated by the model. To learn these classes, we use a combination of a simple clustering algorithm and a gradient-descent based algorithm that learns a single layer of a convolutional

Questions ??

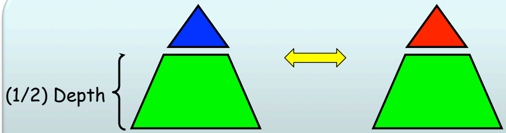
Thank you

Proof Ideas

- Upper bound for small mutation:
  1) distance estimation
  2) reconstruct one (or a few) level(s)
  3) infer sequences at roots



Lower bound for high mutation:



need $k \propto \exp\left(\frac{1}{2}\text{Depth}\right)$

The formal Model

- Let $T = (V, E)$ be a d-ary tree with h levels.
- To each node $v \in V$ associate a representation $\sigma(v) \in [q]^k$
- The process $(\sigma(v))_{v \in V}$ is a Markov Chain on the tree.
- Let $L(v)$ denote the set of labels of v.
- *Assume*: the set of nodes with label ℓ are all the nodes below a certain node v_ℓ.
- Semi-supervised inference problem: Given
  1. Labeled data: $[(\sigma(v), L(v)) : v \in D_L]$ and
  2. Unlabelled data $[(\sigma(v)) : v \in D_U]$ where $D_L \cup D_U$ are the leaves of the tree.
- Find $L(v)$ for all (most) $v \in D_U$.

Examples

- Let $L(v) \in$ Dog, Cat, Labrador etc.
- Let $\sigma(v)$ be the DNA sequence of leaf v, or
- Let $\sigma(v)$ be an image of leaf v etc.

The Markov Chain - Easy Version

- Representations evolve from one layer to next via:
  1. If $v \to u$, the given $\sigma(v)$, it holds for all $1 \leq i \leq k$ independently that
  2. $\sigma(u)_i = \sigma(v)_i B(v) + (1 - B(v))U(v)$ where $B(v)$ are i.i.d. Bernoulli θ and $U(v)$ are i.i.d $U[q]$.
- This is a standard model of evolution in biology.

The Markov Chain - Hard Version

- Representations evolve from one layer to next via:
  1. If $v \to u$, the given $\sigma(v)$, for all $1 \leq i \leq k$ independently set
  2. $\tau_i = \sigma(v)_i B(v) + (1 - B(v))U(v)$ where $B(v)$ are i.i.d. Bernoulli θ and $U(v)$ are i.i.d uniform.
  3. $$(\sigma(v)_{2i-1}, \sigma(v)_{2i}) = P\left(\tau_{\Sigma(2i-1)}, \tau_{\Sigma(2i)} \right)$$
  4. where P is a permutation on $[q]^2$ that depends only on the level and
  5. Σ is a permutation of the k positions that depends on the level h'
  6. Major example k is a power of 2 and Σ is the involution that exchanges a and $a \oplus 2^{h'}$.
  7. Models interaction between features as well as the non canonical nature of representations.

From one object to many

- Tree of objects.
- Sister objects share all representations but the last level.
- Cousins share all representations but last two levels etc.
- E.G.: Top node- mammals, a lower node: dog etc.

The Inference Problem

- Data: two collections of objects:

$$\Big(\sigma^h(u) : u \in A\Big), \quad \Big(\Big(\sigma^h(u), L(u)\Big) : u \in B\Big)$$

  where $L(u)$ is the label of u (e.g. dog, cat, etc.)
- Goal: Find $L(u)$ for $u \in B$.
- This is a *semi-supervised* learning problem.

(Technical) Assumptions

- The tree of objects is a d-ary tree of h levels.
- For any label a:
  - The set of nodes labelled by ℓ consists of all nodes descending from some node v_ℓ.
  - There are $u_1, u_2 \in B$ whose most common ancestor is v_ℓ such that $L(u_1) = L(u_2) = \ell$.
- $\implies$ if location of leaves in tree is known, can label A correctly.

Main Questions

- When can we label leaves correctly?
- Which algorithm can do so?
- Do they have to be “deep”?

What is a shallow algorithm?

- A shallow algorithm is an algorithm that cannot use interaction between the features of the labelled data. More formally:
- Let A denote the unlabelled data and B denote the labelled data.
- The input to the shallow algorithm is:

$$\Big(\sigma^h(u) : u \in A\Big),$$

$$\Big(n_\ell(j,a) : a, 1 \leq j \leq k\Big), \quad n_\ell(j,a) := \Big|\{v : v \in B, L(v) = \ell, \sigma_j^v = a\}$$

Main Results and Conjectures

- Assume $q \to \infty$.

Main Results and Conjectures

- Assume $q \to \infty$.
- Positive: $\theta > b^{-1} \implies$ tree recovery and correct labelling.

Main Results and Conjectures

- Assume $q \to \infty$.
- Positive: $\theta > b^{-1} \implies$ tree recovery and correct labelling.
- Negative: $\theta < b^{-1/2} \implies$ *shallow algorithms* fail.

Main Results and Conjectures

- Assume $q \to \infty$.
- Positive: $\theta > b^{-1} \implies$ tree recovery and correct labelling.
- Negative: $\theta < b^{-1/2} \implies$ *shallow algorithms* fail.
- Conjecture: $\theta < 1 - \exp(-Ch) \implies$ *shallow algorithms* fail.