Game Theory and ML

Start Recording!

Game Theory and ML

IFT 6756 --- Winter 2021 --- Gauthier Gidel

gauthier.gidel@umontreal.ca

01
ABOUT THE COURSE

02
WHY GAMES?

03
STREET FIGHTING MATHS

04
Why do we care about proofs?

1. ABOUT THIS COURSE

Game Theory and Machine Learning

- Taught in English
- You can always ask questions in French.
- Evaluation: 100% Project

Game Theory and Machine Learning

- Project:
  - Validation of the projects with a mid-term proposal/abstract. (Beginning of March)
  - Project presentations at the end of the semester (Talk) + Written report.
  - Three kind of projects:
    - Paper critic. (Less risky)
    - “Open” projects among the list. (More risky)
    - Propose your own project. (More ambitious though I’ll take it into account)

Game Theory and Machine Learning

- All the courses are recorded.
- I will put a short(er) video to watch **before** the course.
- Fill the google forms.
- Go the the TEAMS group and ask questions there.

What is it about?

- Game Theory: *the study of mathematical models of strategic interaction among rational decision-makers.* [Myerson, 1991]

What is it about?

- Game Theory: *the study of mathematical models of strategic* ***interaction*** *among rational decision-makers.* [Myerson, 1991]

What is it about?

- Game Theory: *the study of mathematical models of strategic **interaction** among **rational** decision-makers.* [Myerson, 1991]

What is it about?

- Game Theory: *the study of mathematical models of strategic **interaction** among ~~rational~~ **decision-makers**.* [Myerson, 1991]

  + Machine Learning

What is it about?

- Game Theory: *the study of mathematical models of strategic* ***interaction*** *among* ~~*rational*~~ ***decision-makers***. [Myerson, 1991]

+ Machine Learning

How to learn these strategic behaviors!!!

We will cover
(biased by my Own interest...)

- Standard Game Theory
- Standard ML
- GANs
- Optimization
- Optimization of Games
- Some Aspect of Multi-Agent RL

We will cover (biased by my Own interest...)

- Standard Game Theory
- Standard ML
- GANs
- Optimization
- Optimization of Games
- Some Aspect of Multi-Agent RL

Importance of the theory.
(See end of this course)

We will cover (biased by my Own interest...)

Importance of the theory.
(See end of this course)

- Standard Game Theory
- Standard ML
- GANs
- Optimization
- Optimization of Games
- Some Aspect of Multi-Agent RL

"Supposed" to be self contained.
(Undergrad math background "required"
- Linear Algebra
- Functional analysis)

Who is this Class for?

- Anyone who is interested in ML and Games.
- Target audience:
  - Graduate students with ML background.
- Minimal math background is important to follow some parts of the course.

Required Background

- Exercices (not graded).
- Prove some elementary results that will be used in the course.
- It is the basic knowledge to:
  - Read ML papers (related to that topic)
  - Attend ML conferences (related to that topic too)
- If exercices are too easy: GREAT (work on your project)
- If too hard: some pointers are provided.

What you can expect

- A good grade? (Yes if you ask questions put some work on your project)
- Learn new things!!!
- Answers to your questions.
- 30-40 min Video of the Lectures 3-4 days before the class

What I expect from you

- Watch the Video.
- Do (try) the Exercices.
- Ask Questions!
- A serious project.

To be Active and make mistakes

What I expect from you

- Watch the Video.
- Do (try) the Exercices.
- Ask Questions!
- A serious project.

To be Active and make mistakes
(If you don't I failed)

II. Why Games?

(Largely inspired from the Great NeurIPS tutorial on learning dynamics by Marta Garnelo, Wojciech Czarnecki and David Balduzzi,) If you have a **large dataset,** and you train **a very big neural network**, then **success is guaranteed!**

Ilya Sutskever

Deep learning is just **glorified** curve fitting

Quora User 20034

Optimization!

Deep learning is just **glorified** curve fitting

Quora User 20034

Optimization!

Deep learning is just **glorified** curve fitting

Quora User 20034

“All the impressive achievements of deep learning amount to just curve fitting,”

Judea Pearl (Turing award 2011)

If you have a ~~**large dataset,** and you train **a very big neural network**~~, then **success is guaranteed!**

Ilya Sutskever

If you have the **right objective** and you have enough **capacity and compute**, then **success is guaranteed!**

(improved) Ilya Sutskever

If you have the **right objective** and you have enough **capacity and compute**, then **success is guaranteed!**

(improved) Ilya Sutskever

- Optimizees: “Things that optimize” (architecture, algorithms)
- Objectives : “Target of the optimzation” (data+loss, env+reward)

Getting the Right Objective

Even assuming that the rest is solved (dream world)

- How to **construc**t objectives?
- How to **evaluate** objectives?
- How to **combine** objectives?

Constructing a Task

- Need to curate a **dataset** (MNIST, CIFAR, ImageNet)
- OR Need to build an **environment** (Atari, chess, Go, Starcraft II)

PROBLEMS

Too costly

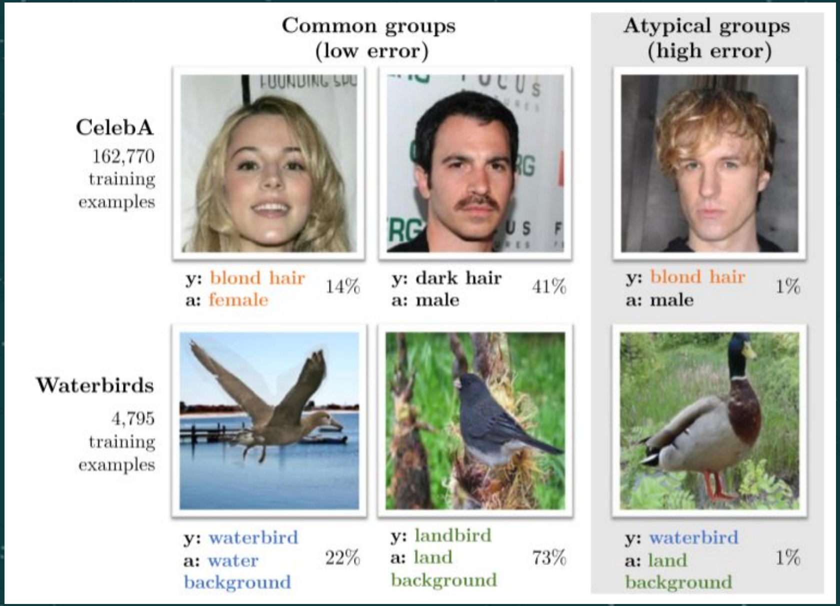
curating datasets or building environments is labour intensive)

[Beyer et al. 2020]

Too simple

Spurious correlations

[Sagawa et al. 2020]



Too many

With a lot of task, easier to cheat.

Source: deepmind.com

Tasks are the main Bottleneck of Progress

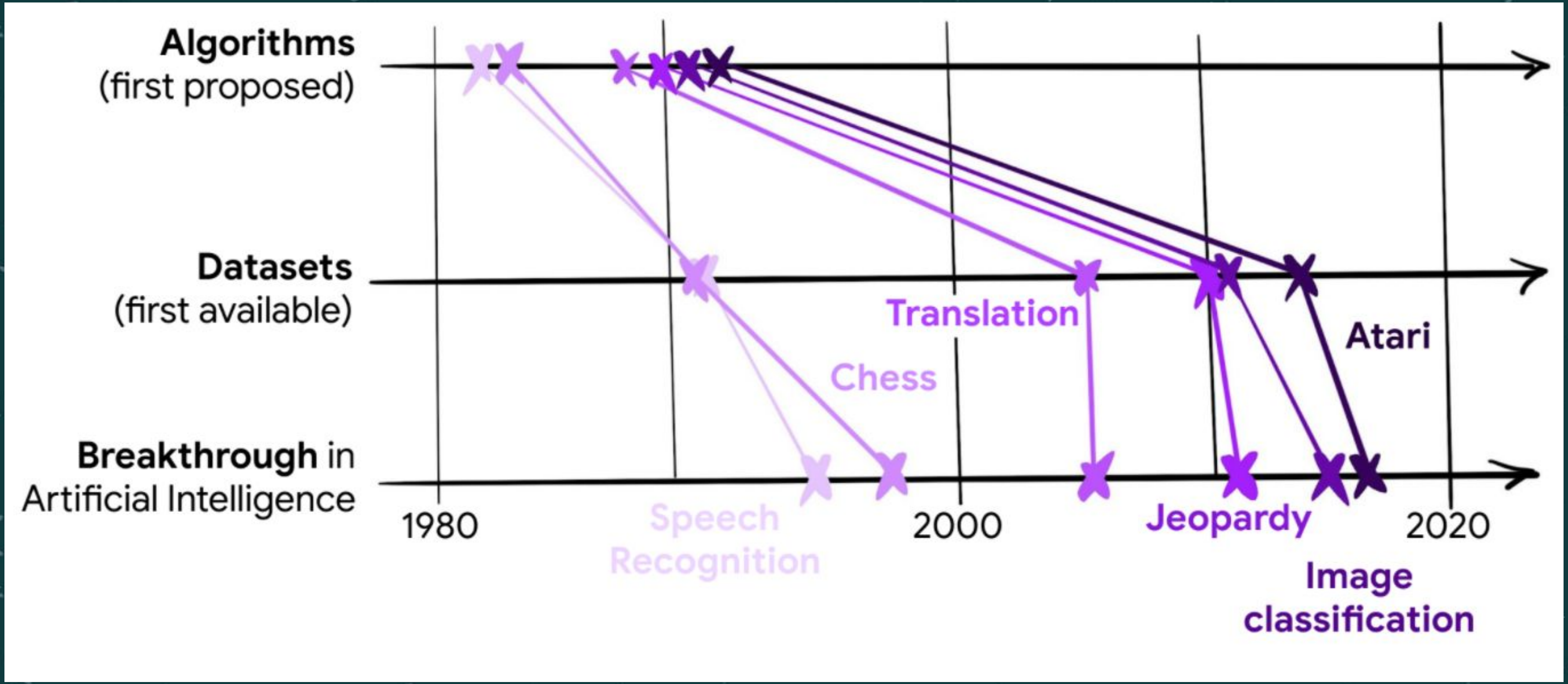


Figure from NeurIPS tutorial on learning dynamics by Marta Garnelo, Wojciech Czarnecki and David Balduzzi

What about Guarantees???

- **Theoretical CS:**
  Algorithms have specific guarantees. (e.g. sorting,...)

- **Machine Learning:**
  We indirectly specify the task via an objective.

- **Learning Theory:**
  Formal "generalization" guarantees based on *unrealistic assumptions (e.g. train set == test set,* i.i.d samples,...*)*

How Tasks are Evaluated?

- Why ImageNet? MNIST? (why vision?)
- Science is a social thing.
- No formal criteria to select the “most interesting” tasks. c

How Tasks are Combined?

	Task 1	Task 2	Task 3	AVG	Rank
Agent 1	89	93	76	**86**	1
Agent 2	85	85	85	**85**	2
Agent 3	79	74	99	**84**	3

Table from NeurIPS tutorial on learning dynamics by Marta Garnelo, Wojciech Czarnecki and David Balduzzi

How Tasks are Combined?

	Task 1	Task 2	Task 3	Task 3'	AVG	Rank
Agent 1	89	93	76	77	**83.75**	**3**
Agent 2	85	85	85	84	**84.75**	**2**
Agent 3	79	74	99	98	**87.5**	**1**

Averaging is a dangerous game.

Table from NeurIPS tutorial on learning dynamics by Marta Garnelo, Wojciech Czarnecki and David Balduzzi

Evolution is NOT (single objective) Optimization

- Fitness is not a function of a single being.

  Survival of the strongest.

- Life is not a suite of tasks.

  School is..., but in real life we do not average.

- Rewards are crazily sparse.

  (YOLO)

Learning Objectives

- Learning

  Do not want to *hand-craft behavior.*

  Catch: Learning from examples but lose behavioral guarantees.

- Learning Representations

  Do not want to *hand-craft features.*

  Catch: Lose optimization guarantees (non-convex optim)

- Learning Losses.

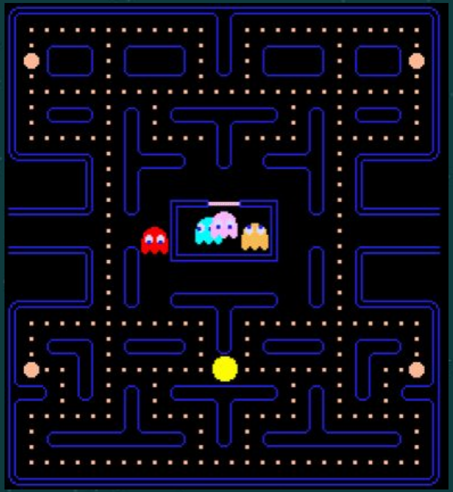
  Do not want to *hand-craft* tasks.

  Catch: Learn against another agent. Tasks: being the fittest

  (e.g. GANs see Courses 4 to 8)

Learning Objectives

Single player:



Hand-crafted notion of performance

Multi-player:

Very simple notion of performance
The complexity of the task depends on the **opponent(s)**

Achieving super-human performance in Chess has been long standing challenge

[Shannon 1950]

Programming a computer for playing chess.

[Samuel 1959]

Some studies in machine learning using the game of checkers

Deep Blue (1996)

Matthew Pritchett

Deeper Blue (1997)

Photo: EPA

[Campbell et al. 2002]

Research paper on Deep Blue

Beyond Chess, achieving super-human performance in multi-player games are great challenges

Go

[Silver et al. 2016]
(Picture from DeepMind's blog post)

Dota 2

[OpenAI et al. 2019]
(Picture from OpenAI's Blog post)

Poker

[Brown and Sandholm 2019]
(Picture from FAIR's Blog post)

Starcraft II

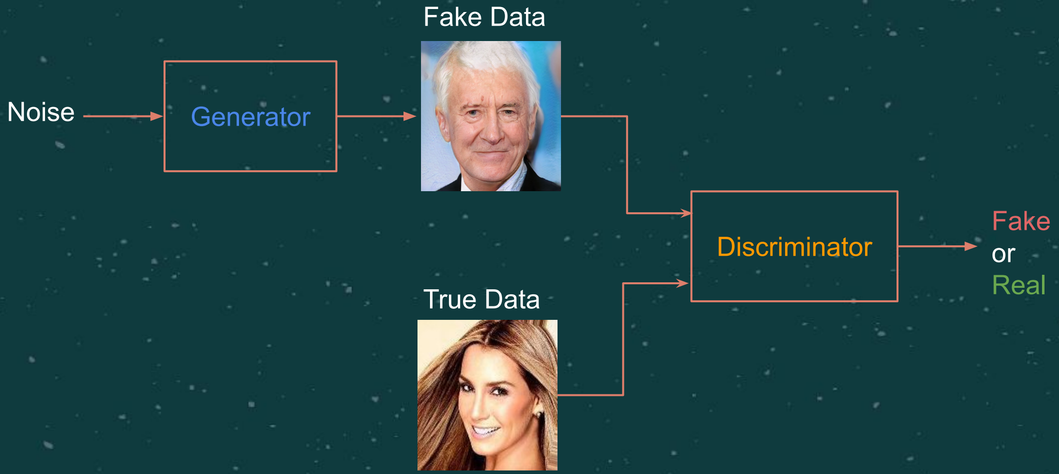
[Vinyals et al. 2019]
(Picture from DeepMind's Blog post)

Games specifically designed for Machine learning purposes

For Generative modeling:

Generative Adversarial Networks
[Goodfellow et al. 2014]

Picture: [Wu et al. 2020]



Games are a great tool to learn complex notions

III. Street Fighting Maths

(title due to Sanjoy Mahajan and Ryan O'Donnell)

What is this sequence?

3, 10, 5, 16, 8, ...

What is this sequence?

3, 10, 5, 16, 8, ...

Ask www.oeis.org

Is this true?

$$|1 + \lambda + \lambda^2| \leq 1 , \quad \forall \lambda \quad \text{s.t.} \quad \Re(\lambda) > 0$$

$$\sqrt{1+x} \geq 1 + \frac{1}{2}x - \frac{1}{8}x^2 \quad \forall x \geq \cancel{-1}\ 0$$

Ask Wolfram!

www.wolframalpha.com

What are the Chebyshev Polynomials of Second kind ?

(and how do you spell Chebyshev???)

Ask Wikipedia!!!

Street Fighting Maths in Practice

- Let us consider a unitary polynomial of degree n: $P_n(X) = X^n + \dots$

  What is the smallest value we can get for $\max_{x \in [-1,1]} |P_n(x)|$?

- Overall it is a Minimax problem: (appears in optimization of games)

$$\min_{P_n \in \mathcal{P}_n} \max_{x \in [-1,1]} |P_n(x)|$$

- How to approach this?

A first Minimax problem

Ways to solves this:

- Googling
- Ask on stackexchange
- Prove it straight (good luck)
- Use Street Fighting Maths (What we will do now)

A first Minimax problem

- How do we start?
- Let us try small values of n.
- $n = 0$:
- $n = 1$:

A first Minimax problem

- How do we start?
- Let us try small values of n.
- n = 0 : $P_0(x) = 1$ $\longrightarrow$ OK
- n = 1 :
- n larger:

A first Minimax problem

- How do we start?
- Let us try small values of n.
- n = 0 : $P_0(x) = 1$ $\longrightarrow$ OK
- n = 1 : $P_1(x) = X + a$ $\longrightarrow$ $\min_a \max_{x \in [-1,1]} |x + a|$ $\longrightarrow$ $a = 0$
- n larger:

A first Minimax problem

- How do we start?
- Let us try small values of n.
- n = 0 : $P_0(x) = 1$ ⟶ OK
- n = 1 : $P_1(x) = X + a$ ⟶ $\min_{a} \max_{x\in[-1,1]} |x+a|$ ⟶ $a = 0$
- n larger:

$$\min_{a_0,\dots,a_{n-1}} \max_{x\in[-1,1]} |a_0 + \ldots + a_{n-1}x^{n-1} + x^n|$$

A first Minimax problem

1. How do we solve this?

$$\min_{a_0,\ldots,a_{n-1}} \max_{x\in[-1,1]} |a_0 + \ldots + a_{n-1}x^{n-1} + x^n|$$

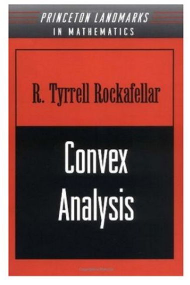
2. Non-convex in x.... (we want global max) .

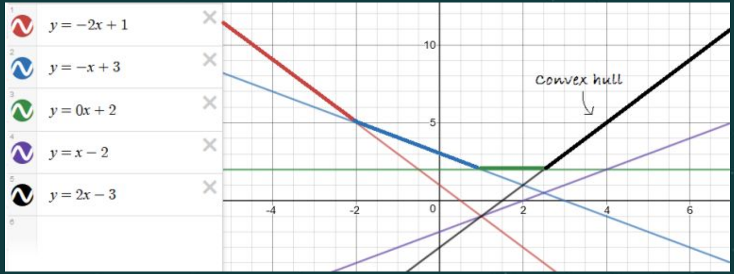
A first Minimax problem

1. How do we solve this?

$$\min_{a_0,\ldots,a_{n-1}} \max_{x\in[-1,1]} |a_0 + \ldots + a_{n-1}x^{n-1} + x^n|$$

2. Non-convex in x.... (we want global max) .
3. Idea: I know some stuffs about convex optimization (see Jamboard)





https://codeforces.com/blog/entry/63823

A first Minimax problem

```python
n_steps = 4000
print(f"d = {0}: Poly: X^0")
for d in range(1,6):
  eta = .1
  a = np.ones(d)
  for i in range(n_steps):
    # loss = get_loss(a)
    g = grad(get_loss)(a)
    a -= eta * g / np.sqrt(i+1)

  poly_str = f"X^{d}"
  for i in range(d-1,0,-1):
    poly_str = poly_str + f" + {a[i]} X^{i}"
  print(f"d = {d}: Poly: "+ poly_str)
```

```
d = 0: Poly: X^0
d = 1: Poly: X^1
d = 2: Poly: X^2 + -9.674113243818283e-08 X^1
d = 3: Poly: X^3 + 0.00020798365585505962 X^2 + -0.7506171464920044 X^1
d = 4: Poly: X^4 + 0.0003095673746429384 X^3 + -1.0005478858947754 X^2 + 0.0011363952653482556 X^1
d = 5: Poly: X^5 + 0.00010893004946410656 X^4 + -1.25143301486969 X^3 + 0.0012422415893524885 X^2 + 0.31438684463500977 X^1
```

A first Minimax problem

- Summary, if we look at $2^{n-1}P_n(x)$:
  - N=0: 1
  - N =1: X
  - N=2: $2X^2 - 1$
  - N=3: $4X^3 - 3X$
  - N=4: $8X^4 - 8X^2 + 1$

A first Minimax problem

- Summary, if we look at $2^{n-1}P_n(x)$:
  - N=0: 1
  - N =1: X
  - N=2: $2X^2$ - 1
  - N=3: $4X^3$ - 3X
  - N=4: $8X^4$ - $8X^2$ + 1
- What if we search for, 1,1,2,-1,4,-3,8,-8,1 ? on www.oeis.org

A first Minimax problem

The OEIS Foundation is supported by donations from users of the OEIS and by a grant from the Simons Foundation.

THE ON-LINE ENCYCLOPEDIA OF INTEGER SEQUENCES ®

founded in 1964 by N. J. A. Sloane

Thanks to everyone who made a donation during our annual appeal!
To see the list of donors, or make a donation, see the OEIS Foundation home page.

1,1,2,-1,4,-3,8,-8 Search Hints

(Greetings from The On-Line Encyclopedia of Integer Sequences!)

Search: **seq:1,1,2,-1,4,-3,8,-8**

Displaying 1-2 of 2 results found. page 1

Sort: relevance | references | number | modified | created Format: long | short | data

A028297 Coefficients of Chebyshev polynomials of the first kind: triangle of coefficients in expansion of cos(n*x) in descending powers of cos(x). +30 25

1, 1, 2, -1, 4, -3, 8, -8, 1, 16, -20, 5, 32, -48, 18, -1, 64, -112, 56, -7, 128, -256, 160, -32, 1, 256, -576, 432, -120, 9, 512, -1280, 1120, -400, 50, -1, 1024, -2816, 2816, -1232, 220, -11, 2048, -6144, 6912, -3584, 840, -72, 1, 4096, -13312, 16640, -9984 (list; graph; refs; listen; history; text; internal format)

OFFSET 0,3

COMMENTS

```
Rows are of lengths 1, 1, 2, 2, 3, 3, ... (A008619).
This triangle is generated from A118800 by shifting down columns to allow for (1,
  1, 2, 2, 3, 3, ...) terms in each row. - Gary W. Adamson, Dec 16 2007
Unsigned triangle = A034839 * A007318. - Gary W. Adamson, Nov 28 2008
Triangle, with zeros omitted, given by (1, 1, 0, 0, 0, 0, 0, 0, 0, ...) DELTA (0,
  -1, 1, 0, 0, 0, 0, 0, 0, ...) where DELTA is the operator defined in A084938. -
  Philippe Deléham, Dec 16 2011
From Wolfdieter Lang, Aug 02 2014: (Start)
This irregular triangle is the row reversed version of the Chebyshev T-triangle
  A053120 given by A039991 with vanishing odd-indexed columns removed.
If zeros are appended in each row n >= 1, in order to obtain a regular triangle
  (see the Philippe Deléham comment, g.f. and example) this becomes the Riordan
  triangle (1-x)/(1-2*x), -x^2/(1-2*x). See also the unsigned version A201701 of
  this regular triangle.
(End)
Apparently, unsigned diagonals of this array are rows of A200139. - Tom Copeland,
  Oct 11 2014
It appears that the coefficients are generated by the following: Let SM_k = Sum(
```

A first Minimax problem

After some googling (with the right keywords)....

2

EXTREMAL PROPERTIES

One of the most remarkable properties of the Chebyshev polynomial, $T_n(x)$, is that $\tilde{T}_n(x)$ (the Chebyshev polynomial normalized so that its leading coefficient is 1) has the smallest maximum absolute value on I: $[-1, 1]$ among all $p(x) = x^n + a_{n-1}x^{n-1} + \cdots + a_0$ [cf. (1.109)] (This property is one basis for the wide utility of the Chebyshev polynomials in numerical analysis, a topic to which we turn in Chapter 3.) Let us begin by proving this fact. We recall that if $g(x)$ is continuous on I

$$\|g\| = \max_{-1 \leqslant x \leqslant 1} |g(x)|.$$

Theorem 2.1. If $p(x) = x^n + a_{n-1}x^{n-1} + \cdots + a_0$, then

$$\|p\| \geqslant \|\tilde{T}_n\| = \begin{cases} 2^{1-n}, & n > 0, \\ 1, & n = 0, \end{cases}$$

with equality only if $p = \tilde{T}_n$.

IV. Why do we care about proofs (and rigor)?

A Universal Optimization Algorithm

A Universal Optimization Algorithm

- Beginning of the talk: We assumed Optimization was “solved”
- What does it mean?

A Universal Optimization Algorithm

- Desired properties for a "perfect" algorithm:
  - Converges to the solution.
  - Works for convex function.
  - Works for non-smooth fuctions (e.g. ReLU)
  - Handle discrete variables.
  - Simple
  - Handle constraints and non convex domains.
  - Can solve nonconvex-nonconcave minimax problems

Too good to be true?
Where is the Trick?

A Universal Optimization Algorithm

- I want to solve: $\min_{x \in X \subset \mathbb{R}^d} f(x)$

[See Jamboard]

A Universal Optimization Algorithm

- Desired properties for a "perfect" algorithm:
  - Converges to the solution.
  - Works for convex function.
  - Works for non-smooth fuctions (e.g. ReLU)
  - Handle discrete variables.
  - Simple
  - Handle constraints and non convex domains.
  - Can solve nonconvex-nonconcave minimax problems
  - Reasonable convergence rates!!!!! (i.e. being practical)

A Universal Optimization Algorithm

- Desired properties for a “perfect” algorithm:
  - Converges to the solution.
  - Works for convex function.
  - Works for non-smooth fuctions (e.g. ReLU)
  - Handle discrete variables.
  - Simple
  - Handle constraints and non convex domains.
  - Can solve nonconvex-nonconcave minimax problems
  - Reasonable convergence rates!!!!! (i.e. being practical)
  - Theory of ML: Try to capture that practical aspect. (sample complexity and Convergence rates!!!)

Thanks everyone!

Do no forget to fill the Google form!