PROBABILISTIC MODELS IN MODERN AI

1. Foundations: What AI Means

- What is the goal of AI? To construct dynamical systems that will process information, appropriately reacting to inputs, and to have some meta-program explain how this dynamical system will be shaped from an environment (either static, with data, or dynamic, or both). They will construct a representation of the data that allows for some downstream tasks that are not describable using usual programs
- Modern AI is about a search in the space of programs, and the identification of those that will do well.

1.1. Goals of Modern AI.

- A central long-term goal for AI is to create programs that perform tasks that require intelligence, following instructions we cannot explicitly write down.
- Modern AI seeks to build programs that autonomously construct representations of data and act on them under uncertainty.
- These programs must be able to play games, interact with streams of information and do clever things.
- In the end, these programs are just a family of functions. The thing is that are quite many functions out there; how do we find the right ones is not completely trivial.
- To find the right functions, we need an environment: either static, or dynamic; the earlier is already the source of many exciting things, while the latter is obviously at the heart of current developments.
- It is not very important to define intelligence, but for us intelligence is the ability to treat streams of information and to find interesting things/patterns in them, with regards to some goals, which can be explicit or implicit, known in advance or unforeseen.
- Some central tasks that drive the construction of intelligent structures are prediction, compression, denoising; to perform those, a training algorithm will typically need to find patterns in data.
- What are the branches of modern AI?
  - Symbolic/Knowledge-Based
    - Search, planning
    - Kernel methods
  - Machine Learning
    - Supervised: we have a training dataset of data points with labels, and we try to fit the labels
    - Unsupervised: we have a training dataset made of samples, and we try to model the dataset
    - Self-Supervised: we have a training set where data points can be used as labels for others

    - Reinforcement Learning: we have an environment that responds to actions
    - Miscellaneous: semi-supervised learning, transfer learning, online learning, meta-learning
- Some other (more or less useful) dichotomies:
  - Deterministic vs probabilistic algorithms: is the output of the AI system deterministic or random? Most neural networks output deterministic outputs, but these can be fed into (or be fed to) some random variables, to generate random outputs.
  - Generative vs discriminative algorithms: we can either sample from some data, or try to infer parameters behind the data.
  - Static versus dynamic environment: a dataset is e.g. a static environment (we don't choose what we see), while a game is e.g. a dynamic environment
- Information theory is the foundation of what we do in modern AI and how we work with this course. Information theory is about what can be done theoretically (as opposed to algorithmically or practically) with information at our disposal. Information theory gives a baseline.
- Modern AI is in my view at the intersection of information theory and practical optimization. Information theory is in some sense the limit of what we can do, and it also suggests some general optimization tasks we can perform. From there, we obtain objects that can be transformed into e.g. intelligent agents.

1.2. **Modern AI in a Nutshell.**

- The most exciting general tasks devised by information theory are prediction, compression, denoising; and they turn out to be the most exciting ones to train AI models
- These three tasks allow one to transform to formulate optimization problem to train e.g. neural networks: we optimize on a space of functions in regard of such tasks. (generally speaking, optimization problems can be applied to labeled or unlabeled data or to environments). And this optimization process yields models with some capabilities. Then there is the question of how we leverage these capabilities to do things.

1.3. **What this course will be about.**

- We will start with the study of prediction and lossless compression as information-theoretic tasks.
- Then we will study neural networks, which provide a means to optimize.
- Later, we will discuss lossy compression and denoising. This will lead us to diffusion models.
- We will finish by discussing reinforcement learning, causality, and games.
- This is will lead us to the 'universal AGI' ideas that are at the foundation of the current wave of the field.

2. Prediction and Lossless Compression

- We start with prediction and lossless compression, two fundamentally related information-theoretic tasks. These lead (once we can optimize properly) to self-supervised learning, including LLMs, other things.

2.1. **Machine Learning and Prediction.**

- Intelligence has often been likened to the ability to predict, not only for machine learning, but also for animal and human intelligence. For instance, it is clearly one of the mechanism by which neurons learn, and 'predicting the next symbol in a sequence' is emblematic of IQ tests.
- In general, if we are to optimize something (as we will discuss with neural networks) we need a clear, universal objective, that can be attached to data. If the data has a time structure or a space structure, we can naturally formulate prediction tasks. The prediction task is extremely deep, as performing it well entails a good model of the world that generates the data. Next-token prediction, infilling, backward prediction are all part of that galaxy of tasks.
- Maybe we can start with an example: if we compute the first terms of a mathematical sequence... if we give 6.2831853 and we ask what is the next digit, how do we answer this? It would make sense to say that the answer is 0... why? This is not necessarily the simplest task, but an explanation here would be that 2π is a good summary of the data, and it is a low-complexity one; again, as we will see, it corresponds to compressing the data.
- For textual data, supervised learning can be viewed as particular case of prediction, if we make the labels follow the data, e.g. if we write "$x_1 :: y_1$ $x_2 :: y_2$ $x_3 :: y_3$ ...", and then we provide a $x ::$ and ask the model to predict after, it should predict y. This is in fact how we would use a foundation LLM for many prediction tasks.
- There are many closely related tasks, in particular denoising, which is also at the heart of information theory: if we were to noise some digits of 2π, an intelligent algorithm should still be able to denoise them, based on compression ideas, for instance. But for now, it is good to start with prediction, as it is conceptually simpler and it yields some of the very best results in practice.

2.2. **Scoring Rules.**

- Note: the introduction of scoring rules is a bit unusual as a treatment of AI. We follow this road as a means to get into information theory without knowing information theory a priori. This is based on questions and results that were already available at least four hundreds years ago (Pascal and Huygens talked about it, the Bernoulli worked on it), but were maybe not pushed later. The theory of probability was in large part developed in the context of gambling, and prediction and gambling are very obviously related.
- What is a prediction for a random variable? Assume for simplicity and concreteness that it can only take a finite number of values. Predicting consists in delivering probabilities for the various outcomes of that variable, as available to an agent based on their information. The goal of a scoring rule is to reward an agent outputting predictions based on their predictions and on the outcome.
- Of course, the outcome of a random variable is a single value, and this is the only feedback about how good a prediction was going to be. We can still try to make the rewards so as to *elicit* the right reward, i.e. to make

it so that the agent maximizes their reward when they give the correct probabilities.

- So, let us formalize this: an agent outputs probabilities $\vec{\pi} = (\pi_1, \dots, \pi_n)$ for the possible values $\{1, \dots, n\}$ that a random variable could take. The space of possible outputs is the simplex $\Delta_n \subset \mathbb{R}^n$ defined as the set of n-dimensional vectors with nonnegative entries summing up to 1.
- Then a scoring rule s takes as input $\vec{\pi}$ and the actual (random) outcome of the observed variable i to yield a reward $s(\vec{\pi}, i)$.
- A scoring rule is called *proper* if the only maximizer of the expectation

$$\mathbb{E}_{\vec{p}}[s(\vec{\pi}, \cdot)] = \sum_{i=1}^{n} p_i s(\pi_i, i),$$

assuming 'true' probabilites $p_1, \dots, p_n$ for the outcomes, is when $\vec{\pi} = \vec{p}$. The 'subjective' view of this question is that as much as agents don't know the true probabilities, they are incentivized to disclose their own 'beliefs': their goal is to maximize their (perceived) expectation.
- Note that this point of view is very much related to a deep belief at the core of modern AI: subjectivism/bayesianism, which itself is at the heart of information theory. The idea is not really that there is mathematical model of the world that we can define and study mathematically, just that there are models of reality, that there is a feedback, and that we should just update the beliefs based on feedback. Of course, this is more a 'general vision' than anything else: it is not forbidden to assume that there is a more or less correct vision of the world; it is just not to be assumed that it can be defined accurately or accessed in any way.
- Ok... so how do we design proper scoring rules? A naive idea that is bad (but probably used at places) is to reward $s(\vec{\pi}, i) = \pi_i$: this seems reasonable, because if the agent gives higher probabilities to more likely events, their expected score increases... the problem is that they increase beyond the true probabilities: it is easy to see that the optimal strategy is in fact to output $\pi_i = \mathbf{1}_{i=j}$ if $p_j > p_k$ for all $k \neq j$ (exercise: what is the solution if there is no such p_j?).
- What are some examples of proper scoring rules?
  - The quadratic scoring rule, for instance $-(1-\pi_i)^2 - \sum_{k \neq i} \pi_k^2 = -\|\vec{\pi} - \delta_i\|^2$, where δ_i is the i-th canonical basis vector (or 'one-hot' vector) with zero entries at $k \neq j$ and entry 1 at $k = j$.
  - The quartic scoring rule: $4\pi_i^3 - 3\sum_k \pi_k^4$
  - The logarithmic scoring rule (the most beautiful of all): $\log \pi_i$.
- In the exercises, you saw the beautiful characterization of the proper scoring rules in terms of Bregman divergences:
  - Given a smooth strictly convex function $G : \Delta_n \to \mathbb{R}$, we can define a proper scoring rule by

$$s(\vec{\pi}, i) = G(\vec{\pi}) + \langle \delta_i - \vec{\pi}, \nabla G(\vec{\pi}) \rangle$$

  - First note that when taking expectation, for fixed $\vec{p}$, the only 'random quantity' in this expectation that we need to average is $\sum_i p_i \langle \delta_i, \nabla G(\vec{\pi}) \rangle = \langle \vec{p}, \nabla G(\vec{\pi}) \rangle$.

  - Define the Savage representation $S(\vec{p}, \vec{\pi})$ as the expected return $\mathbb{E}_{\vec{p}}[s(\vec{\pi}, \cdot)] = G(\vec{\pi}) + \langle \vec{p} - \vec{\pi}, \nabla G(\vec{\pi}) \rangle$.
  - Why is our scoring rule, now? A small cool twist (related to convex duality): for a fixed $\vec{\pi}$, as a function of $\vec{p}$, we have an affine function that is tangent to the graph of G, and so, since convex functions are always the sup of the affine functions that are tangent to their graphs, $S(\vec{p}, \vec{\pi}) \leq G(\vec{\pi})$ for all $\vec{p} \in \Delta_n$, and in fact we have strict inequality if $\vec{p} \neq \vec{\pi}$.
  - So, now if we fix $\vec{p}$ and optimize $\vec{\pi}$ instead, we find that the best expected return is by taking $\vec{\pi} = \vec{p}$.
  - Conversely, if we have an expected return function that derives from a proper scoring rule, we have, at every fixed $\vec{p}$, some Savage function $S(\vec{p}, \vec{\pi})$; we can define the 'entropy' $H(\vec{p})$ as $\sup_{\vec{\pi}} S(\vec{p}, \vec{\pi})$.
  - Note that $H(\vec{p}) = \langle \vec{p}, s(\vec{p}, \cdot) \rangle$.
  - For every fixed $\vec{\pi}$, we have $S(\vec{p}, \vec{\pi}) \leq H(\vec{p})$, and hence (for fixed $\vec{\pi}$ again), the affine function $S(\cdot, \vec{\pi})$ lies below $H(\cdot)$, and is tangent at $\vec{p} = \vec{\pi}$. Hence, H is convex function of $\vec{p}$ (it is the sup of functions depending on $\vec{\pi}$ lying below it).
  - Hence (assuming differentiability and strict properness), we get that the gradient at $\vec{p} = \vec{\pi}$ of the affine function $S(\cdot, \vec{\pi})$ matches that of $H(\cdot)$.
  - Now, if we evaluate how much we get rewarded in case of outcome i, we collapse the probability to $\vec{p} = \delta_i$, and we get $S(\delta_i, \vec{\pi}) = H(\vec{\pi}) + \langle \nabla H(\vec{\pi}), \delta_i - \vec{p} \rangle$.
  - So, we have found that the scoring rule should be $s(\vec{\pi}, i) = H(\vec{\pi}) + \langle \delta_i - \vec{\pi}, \nabla H(\vec{\pi}) \rangle$
- So (modulo differentiability assumptions), we have found a complete characterization of scoring rules!
- Now, if we ask for locality, i.e. that $s(\vec{\pi}, i)$ only depends on π_i (and not the other π_j's for $j \neq i$), we find that the only solution is $s(\vec{\pi}, i) = \alpha \log \pi_i + \beta$ for some $\alpha > 0, \beta \in \mathbb{R}$.
- As we will see, this leads us to information theory.

2.3. **Compression.**

- In the modern era, if there is a single task that is now understood to be associated with intelligence/understanding, it is compression. This was mostly initiated by Claude Shannon, although the problem was somehow considered before, with e.g. the Morse code representing some attempt to make communication more efficient.
- The ability to synthesize observations about the world down to a few principles is what physics is (abstractly) about, to some extent.
- There is some theoretical ideas as to why the shortest explanation is the best (Occam's Razor)]
- [Entropy, conditional entropy, cross-entropy, mutual information] [Basic properties]
- [Compression] [Summarization] [Synthesis] [Summarize the internet data]
- [Lossless vs Lossy] [Problem of lossy is how to define what we need] [Focus on lossless in this course, because it is easier, and it applies to text]
  - {Huffman, Shannon codings} {McMillan Inequality}

- [Source coding theorem] [Statement: if there is this much information in a channel, you can find a code that compresses it with that many bits] [Proof: one direction, one can use some kind of coding, the other direction is based on the typical sets]
- [Imperfect coding] [Cross-entropy and practical compression] [Kullback Leibler and Gibbs inequality] [Very related to prediction is compression]
- [Classical lossless compression]
  - {LZ Compression} {LZ compresses ergodic things optimally} {Burrows-Wheeler} {Invertibility of BW} {Good properties on text}
- [Arithmetic coding] [Idea]
  - {Optimality proof}
- [Bits back coding] [Idea] [Bits back coding chaining]
- [Asymmetric numeral systems] [Construction and inversion] [What is the point of this]
  - {Proof that construction and inversion work}
- [Now, if we allow for lossy compression, we get into the realm of denoising]

2.4. **Algorithmic Information Theory.**

- [The information theory point of view studies how right we could ever be] [The information theoretic lens looks at data and tells us how much there is to do]
- [There is the question of what we could ever know] [Note that it is not because an information is in principle available that it can be found]
- [Why does the information point of view prove to be more useful?] [It abstracts away the specific algorithm that we use; as opposed to statistics, which tends to focus on constructed quantities] [Information theory is about what could be achieved, theoretically]
- [Kolomogorov-Solomonoff-Chaitin foundations] [Motivation]
- [Kolmogorov complexity definition] [Conditional Kolmogorov complexity] [Universality of Kolmogorov complexity] [Upper bound on conditional complexity by length] [Upper and lower bounds on Kolmogorov complexity] {What we think is unlikely as a random configuration} [The shortest description is upper bounded by the entropy] [The entropy is upper bounded by entropy by Kraft's inequality and source coding]
- [Martin-Löf probability] [Key Concepts] [Examples]
- [Minimum Description Length] [Circuit Length Description]
- [Understand the world generally speaking] [How to process information in a computable way]
- [Information theory comes with a few tasks: compressing and denoising] [Compressing is understanding]
- [Solomonoff induction] [What Solomonoff's induction is about] [Why it is uncomputable]
  - {Kolmogorov Sampler By Donoho}
- [Once we have done this thing, there is the question of what we could do with that] [Later led to AiXi]

2.5. **Where this will go (later in the course).**

- [Universal AGI ideas]
- [AiXi: a theory of RL based upon Solonomoff ideas]

- [Gödel Machines]
- [Now in practice, what do we do is inspired by this vision]

3. Neural Networks

3.1. **Architecture.**

- [The idea is to compose linear maps and nonlinear maps]
- [We take vectors as inputs, and we get vectors as outputs] [To do something interesting, we then need some code that uses that] [For instance, for LLMs, the output would be used to sample tokens]
- [Universal approximation results]
- [Why the architecture is not the only important thing]
- [Early misconceptions about neural networks]

3.2. **Optimization.**

- [Need to fix a selection process with an objective] [The true objective may be different, but we need to select a decent surrogate objective]
- [Saddlepoint problem] [Random initialization] [Gradient descent] [Adam]
- [The specification of a model must involve the optimization task]
- [How long do we run the optimization?]
- [Abstract formulation of gradient with a kernel]
- [Question of large neural networks] [Wrong conjectures]

3.3. **Infinite-Width Limit.**

- [Naive infinite-width limit blows up]
- [Activation kernel scaling regime]
- [Neural tangent kernel description]
- [Law of large numbers]
- [Stability during training]

3.4. **Kernel Description.**

- [The infinite width of neural networks in the kernel regime]
- [The activation kernel]
- [Random features]
- [Gaussian process prior and posterior for kernels]

3.5. **Consequences.**

- [Global minima]
- [Double-descent phenomenon]
- [Generalization]
- [Fine-tuning regime]

4. Large Language Models

4.1. **Auto-Regressive Language Models.**

- [Loss function]
- [Information extraction]

4.2. **LSTMs, GRUs, Transformers, Mamba.**

- [Transformers] [Started with Neural Turing Machines] [Then Bert] [Then GPT]

4.3. **Information Theory.**

4.4. **Compression.**
- [Arithmetic Encoding]

4.5. **Arrows of Time.**

5. Diffusion Models

5.1. **Framework.**

5.2. **Variational Auto-Encoders.**

5.3. **Denoising.**

5.4. **Stochastic Calculus.**

5.5. **Diffusion and Bits Back Coding.**

6. Alternative Paradigms

6.1. **GANs.**

6.2. **Causality.**

6.3. **Bayesian Flow Networks.**

7. Reinforcement Learning: Towards AGI

7.1. **AiXi.**

7.2. **Capability Measures.**

7.3. **Games.**

7.4. **Transfer Learning.**

7.5. **Generality.**

7.6. **Alife Ideas.**