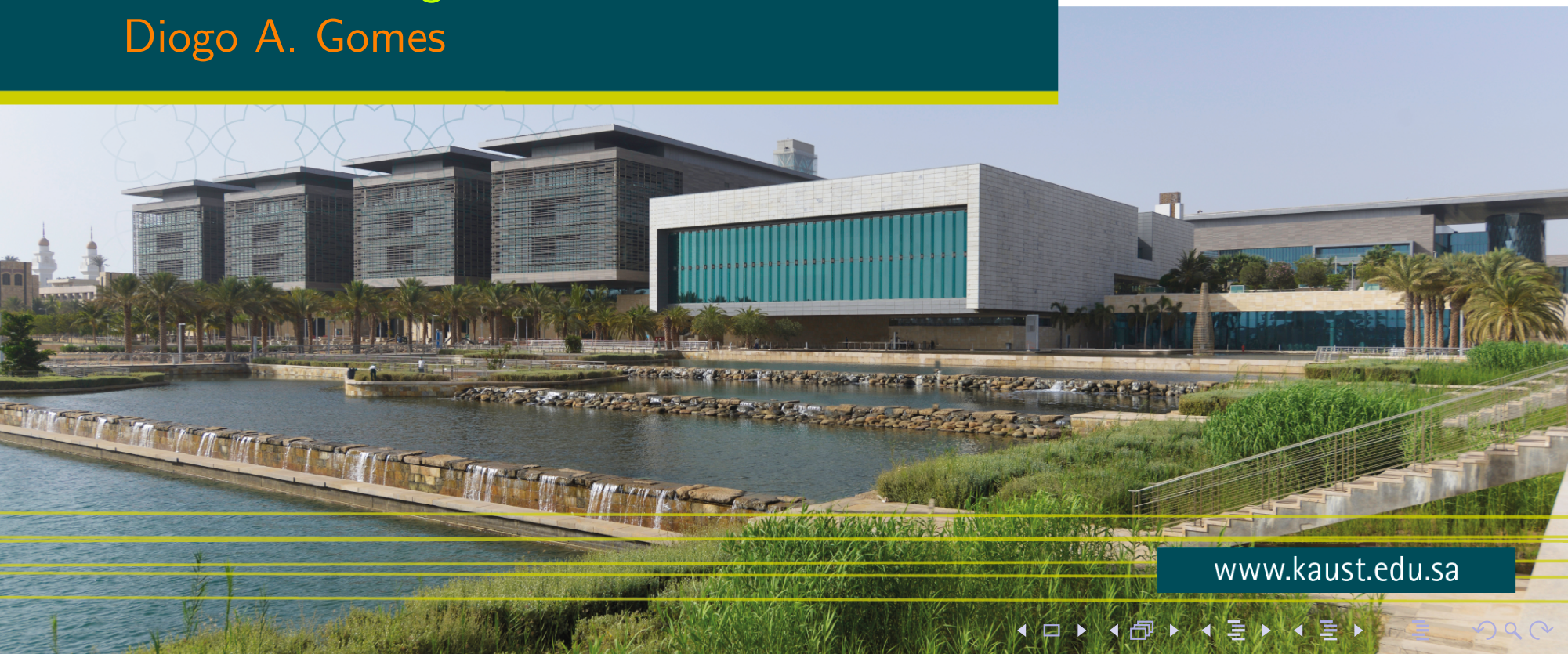
جامعة الملك عبدالله
للعلوم والتقنية
King Abdullah University of
Science and Technology

# Mathematics for AI/ML curriculum design and lessons learned

Diogo A. Gomes



www.kaust.edu.sa

# Mathematics in AI and ML

- Mathematics provides the framework for understanding AI algorithms and their properties.
- Mathematics serves as a unifying language.
- A deep understanding of mathematics can lead to the discovery of new techniques, algorithms, and approaches that push the boundaries of AI and data science.

# Goals

- Expanding beyond basic mathematical methods
- Discuss the role of abstraction in identifying underlying patterns and principles in AI
- Share lessons learned from designing data science MS track and teaching an advanced mathematical course on data science and AI.
- Showcase how AI can contribute to the advancement of mathematical theory and methods
- Stimulate conversation around the evolving role of mathematics in AI.

# Traditional Mathematical Foundations

- Linear algebra
- Probability and statistics
- Optimization methods
- Information theory
- Graph theory

# Opportunities: Integrating AI-Relevant Topics

- Enhance existing courses with AI-relevant content:
  - Incorporate matrix decompositions and basic graph theory in linear algebra courses
  - Introduce Markov chains and Monte Carlo integration in probability and statistics courses
  - Explore multi-step ODE integration and accelerated optimization algorithms in numerical analysis courses
- Benefits:
  - Improve students' understanding of AI applications in various mathematical fields
  - Encourage interdisciplinary thinking and problem-solving skills
  - Prepare students for advanced studies and careers in AI and data science

# Challenges: Balancing Course Content and Teaching Strategies

- Addressing potential issues in curriculum design:
  - Balancing an already packed course syllabus
  - Deciding which topics to remove or condense to make room for AI-relevant content
  - Ensuring students have the necessary prerequisites and foundational knowledge for the new material
- Adapting teaching methodologies and resources:
  - Maintaining student engagement and motivation while introducing advanced topics
  - Utilizing effective teaching strategies to accommodate AI-focused content
  - Continuously updating course materials to keep pace with the rapidly evolving field of AI

# Existing CS courses (a sample)

- **Machine Learning**: Regression, support vector machines, neural networks, with a focus on practical applications and learning theory.
- **Deep Generative Modeling**: Deep Generative Models, Normalizing Flows, Neural ODEs/SDEs, Deep Equilibrium Models, Energy-based Models, and various applications in computer vision, music, and NLP domains.
- **Stochastic Gradient Descent Methods**: Convergence and complexity theory of serial, parallel, and distributed variants of SGD, including accelerated methods and a unified analysis of SGD variants.
- **Federated Learning**: Federated learning, covering supervised machine learning, privacy, distributed and edge computing, optimization, communication compression, and systems.

# Existing AMCS courses (a sample)

- **Numerical Optimization**: Optimality conditions for smooth optimization problems including unconstrained, linear programming, quadratic programming, global optimization, and linearly/non-linearly constrained optimization.
- **Advanced Probability**:measure-theoretic probability, covering probability spaces, random variables, expectations, limit theorems, Radon-Nikodym theorem, conditional expectations, martingales, and applications to Markov chains.
- **Advanced Simulation**: modern stochastic simulation methods, Markov-chain based algorithms like Markov chain Monte Carlo and Sequential Monte Carlo. It covers the development and analysis of various algorithms and their applications in Bayesian inverse problems.
- + Real analysis, functional analysis, numerical linear algebra....

# Key Challenges and Opportunities

- Supplement existing CS courses with mathematical methods directly relevant to ML
- Develop courses on the foundations of ML, focusing on:
  - Approximation properties of deep neural networks
  - Abstract mathematical methods
- Create new courses exploring ML applications, such as: ML methods for solving high-dimensional PDEs

# New courses

- **Math Foundation of Machine Learning** regression (including ridge and Lasso), dimensionality reduction, randomized projection methods, clustering, and graph-based methods, PCA, nearest neighbor classification, k-means, mixture models, and spectral clustering.
- **Deep Learning and Analysis** deep learning techniques, focusing on mathematical and numerical aspects. It targets math students exploring deep learning technology seeking a stronger theoretical foundation. Topics include neural networks, finite element spaces, gradient descent, and multigrid methods.
- **Random PDEs: hierarchical and machine learning approximation** Explores random PDEs, addressing uncertainties in models, and emphasizes efficient numerical solutions using machine learning techniques.

# Abstract Mathematics in AI/ML

- Essential foundations: Traditional mathematical techniques provide critical tools for AI/ML/data science researchers
- Going beyond: A deeper understanding of AI/ML demands the exploration of abstract mathematical concepts
- The role of abstract tools in AI/ML:
  - Uncovering underlying patterns and principles across diverse AI/ML problems and domains
  - Facilitating the creation of generalizable solutions and innovative approaches
  - Connecting AI/ML with other areas of mathematics and fostering interdisciplinary research
- Approaching abstract mathematics in AI/ML:
  - Emphasize the importance of abstract thinking
  - Encourage students to develop their abstract thinking skills
  - Promote collaboration between mathematicians and researchers in AI/ML

# Tentative Syllabus

1. The calculus of variations point of view in supervised learning
2. Properties of random functionals, compactness, and convergence
3. Regularizers, introduction to reproducing kernel Hilbert spaces
4. Laws of large numbers, ergodic theorem
5. Distances on probability measures: Monge-Kantorowich problem, KL divergence, cross entropies
6. Spaces of sequences of random variables, Kolmogorov extension theorem, Markov processes
7. Reinforcement learning: dynamic programming principle, value & policy iterations
8. Topics in optimal control and a dynamical systems view of deep learning
9. Applications to PDEs and a posteriori estimates

# Deep Learning and Calculus of Variations

- Training deep learning models often involves minimizing a loss function, which can be viewed as a functional
- Techniques from calculus of variations can help analyze and optimize the loss functionals in deep learning
- Insights from calculus of variations can improve the stability and convergence properties of deep learning algorithms

# An architecture agnostic approach

Many problems in ML can be phrased a the following problem

- Given an admissible set of maps $\mathcal{A}$
- Find a map $T \in \mathcal{A}$ that minimizes a functional $J(T)$

This is exacly the setting of calculus of variations.

# Traditional calculus of variations approach

In standard calculus of variations, one usually studies this problem as follows

- Show existence of an optimal map - this is usually a combination of the compactness of $\mathcal{A}$ with some continuity of $J$.
- Determine necessary optimality conditions (eg Euler-Lagrange equations).
- Study sufficient conditions, often under convexity assumptions.

# An old problem - new questions

Unfortunately in the applications at hand, $J$ is not accessible. So, the actual setting is

- Given an admissible set of maps $\mathcal{A}$
- Find a map $T_n \in \mathcal{A}$ that minimizes a random functional $J_n(T_n)$

Then, the key question is: is $T_n$ close to optimal for $J$?

# Convergence problem

- $\Gamma$-convergence is the area of mathematics that studies the convergence of functionals $J_n$.
- In statistical learning theory similar problems are often addressed in the contex of VC dimension or Rademacher bounds.

# Empirical risk minimization

Often the setting in ML is the following, we have a underlying probabiliy measure $\mu$ in a product space $X \times Y$, $T : X \to Y$ and we seek to minimize the expected risk

$$J(T) = \int_{X \times Y} c(T(x), y) d\mu(x, y).$$

The empirical risk functional, $J_n$, is as follows, we have a iid sequence $(X_k, Y_k)$ with joint law $\mu$.

$$J_n(T) = \frac{1}{n} \sum c(T(X_k), Y_k).$$

# A simple abstract approach - existence

- Let $\mathcal{A}$ be a compact set on a Banach space.
- Suppose that $J$ is Lipschitz in $\mathcal{A}$

The two preceding conditions imply the existence of a minimizer $\bar{T}$.

# A simple abstract approach - approximation

Assume in addition the following:

- for every $\epsilon$ and every $\delta$, there is $N$ for each $T \in \mathcal{A}$

$$P(J_n(T) - J(T) > \epsilon) < \delta,$$

for all $n > N$.

Because $\mathcal{A}$ is compact it admits an $\epsilon$-cover that is a finite set $T^1 \dots T^d$ such that for any $T$ there exists $i$ such that $\|T - T^i\| < \epsilon$. Fix $\epsilon$ and $\delta$
Let $T_n$ be a minimizer of $J_n$. Then, with probability larger that $1 - d\delta$

$$J(T_n) - J(\bar{T}) \leq C\epsilon$$

# Functional Analysis in Machine Learning

- Compactness in Banach spaces is a key concept in Functional Analysis and has significant implications for Machine Learning (ML) as explained before
- Two notable techniques in ML that draw from Functional Analysis are:
  - Regularization: Adding a regularization term to the objective function, which promotes compactness in the solution space (e.g., $J(T) \to J(T) + |T|_*$)
  - Kernel methods: Transforming the problem of minimizing $J_n$ into a finite-dimensional optimization problem, making use of the Representer theorem and feature space mapping

# Empirical risk minimization - revisited

Often the law of large numbers is used to justify that

$$J_n(T) = \frac{1}{n} \sum c(T(X_k), Y_k) \to J(T)$$

But the (standard) LLN is only valid for independent sequences....

# Convergence of averages of random variables

Suppose we have a sequence of random variables $Z_k$. When is it true that

$$\frac{1}{n}\sum_{k=1}^{n} Z_k$$

converges? and if so, what is the limit?
What about if the sequence is not iid?

# Alternate Convergence Conditions for Averages

- Birkhoff's Ergodic Theorem: Applicable for ergodic processes
- Martingale Convergence Theorem: Applicable for martingales with bounded moments
- Markov Chains: Applicable for finite, irreducible, and aperiodic chains with unique stationary distributions
- Cesàro Mean Convergence: Applicable when the sequence converges in distribution
- de Finetti's Theorem: Applicable for exchangeable sequences of random variables

# Probability Theory in ML

A solid grasp of probability theory is essential for understanding the theoretical foundations of Machine Learning. Convergence of averages of random variables is a key concept:

- ML algorithms estimate unknowns (e.g., parameters, functions) from data, typically treated as random variables.
- Convergence properties reveal algorithm behavior as sample size increases.
- Various learning scenarios demand distinct convergence results (e.g., i.i.d., time-series, exchangeable data).
- Comprehending assumptions and conditions helps select suitable algorithms and tools.
- Identifying limitations informs the development of robust ML algorithms tailored to specific data and applications.

# Spaces of maps

What are typical choices for set $\mathcal{A}$? What are the domain $X$ and range $Y$ of the maps in $\mathcal{A}$? How do we parametrize maps in $\mathcal{A}$?

- Range $Y$: Common choices for $Y$ include discrete sets (classification), vector spaces, and probability measures on a set $K$, with $Y = \mathcal{P}(K)$ (reinforcement learning, language models).
- Parametrization via layer composition: Examples are ReLU + linear layers (finite element spaces) and residual NN (controlled dynamical systems).
- Group equivariance: For a group $G$ acting on $X$ and $Y$, we require $T(g(x)) = g(T(x))$. A prominent instance is the translation group with Convolutional Neural Networks (CNNs), where convolution layers exhibit equivariance to translation, retaining the same translation in input and output maps.

# Metrics in Spaces of Probability Measures

When $Y$ is a normed space a typical distance is the a function of the norm

$$c(T(x), y) = \tilde{c}(\|T(x) - y\|).$$

However, when $Y = \mathcal{P}(K)$, distances on probability measures are needed, such as:

- Monge-Kantorovich distance
- Divergences like KL-divergence

# Impact

- A highly flexible track was developed in the AMCS program, enabling a professional MS offering for ARAMCO and potential future offerings for the Saudi Ministry of Interior employees.
- The new courses attracted students from various programs and had some of the highest enrollment among AMCS courses.

# Abstract Mathematics in AI/ML

- The course only touched the surface of abstract methods in AI/ML, with numerous potential research directions to explore.
- A significant portion of my 20+ years of research is directly linked to reinforcement learning, which may also be true for other mathematicians in various fields.
- Midway through the course, it became evident that we needed to delve deeper into language models and their impact on mathematics, leading us to dedicate the final weeks to exploratory projects.

# Language models

- A language model is a probability measure $P_\theta$ that approximates the empirical probability measure on natural language texts, $P$.
- Often $P_\theta$ can be used to compute conditional probabilities. For example

$$P_\theta(\text{"4"} \mid \text{"What is 2+2? Answer:"})$$

- Often language models are used with a sampler that generates high probability texts.

# Class projects research goals

- Understand the LLM capabilities for mathematical research
- Develop effective strategies for their use
- Propose research directions

# Mathematics paper structure

We can think that a research paper is structured by layers

- Paper overall structure: introduction, assumptions, background material,
- Section specific strucuture: introduction (problem statement, main results...). assumptions (statement of the assumptions, relevance in the paper, examples....)
- Mathematical results organization (theorems, proofs, examples....)
- Paragraph and sentence level structure.
- Spelling

Tools: Spell checkers, grammar checkers, Large language models

# Introduction critic

**Prompt**: Consider the introduction below. Answer the following questions, provide evidence, and offer specific suggestions.

- Did the authors "establish the importance of the field, provide background information, and define terminology if necessary?"
- Did the authors "present the problem and current research focus?"
- Did the authors "explain the motivation for the problem?"
- Did the authors "give an account of prior research in the field and its connection to the current problem?"
- Did the authors "identify a gap in the research, explain the relevance of closing this gap, and describe how their problem contributes to addressing this issue?"
- Did the authors "describe the main result and explain any improvement the main result brings?"
- Did the authors "explain the main techniques used?"

# Introduction critic results

Here are some samples:

- The motivation for the problem is not explicitly mentioned. Suggestion: Explain the motivation for studying the stochastic analog of Aubry-Mather theory.
- The authors do mention prior research and provide citations for it. However, the connection to the current problem could be made clearer.
- The authors do not explicitly mention a gap in the research. Clearly identify the gap in the research and explain how addressing the problem contributes to closing this gap.
- Yes, the authors mention the techniques used, such as Fenchel-Rockafellar duality theorem and viscosity solutions of Hamilton-Jacobi equations.

# Shortcomings and limitations

$$37246823682746378+8236749723467923468 = 82739965471506698 46$$

whereas the right answer is

$$8273995948742179846$$

# Shortcomings and limitations

**Problem:** Show that (0,1) is a closed set.
**Solution:** The set (0,1), its complement is the set $[-\infty, 0) \cup [1, +\infty)$, which is clearly an open set. Therefore, (0,1) is a closed set.

# Progress

**Problem:** Show that (0,1) is a closed set.
**Solution:** To show that the interval (0, 1) is **not** a closed set, we can use the definition of a closed set, which states that a set is closed if it contains all of its limit points. A limit point of a set S is a point x such that every open neighborhood around x contains at least one point from S different from x itself. Now, consider the set (0, 1). It does not include the endpoints 0 and 1. However, both 0 and 1 are limit points of this set, as for any open neighborhood around them, we can always find a point in (0, 1).

# Semantic generation

Can we efficiently generate datasets to train LLM to perform semantic conversion:

- "The sum of 2 and 3 is 5" $\rightarrow$ "The sum of 2 and 3 is Plus[2,5]"
- " A gym charges 50 dollars per month for a basic membership and an additional 30 dollars per month for a personal trainer. If Mike signs up for the basic membership and personal trainer for 3 months, he spends 240 dollars in total." $\rightarrow$ "A gym charges 50 dollars per month for a basic membership and an additional 30 dollars per month for a personal trainer. If Mike signs up for the basic membership and personal trainer for 3 months, he spends Times[3, Plus[50, 30]] dollars in total."

# Successes and Failures of LLMs

- English language: Abundant examples and training data. Multi-step logical reasoning less relevant.
- Coding: Numerous examples, compact language, multi-step reasoning necessary.
- Arithmetic operations: Limited language, fewer examples compared to potential scenarios, multi-step reasoning necessary.
- Mathematical reasoning: Limited examples, diverse language, multi-step logical reasoning essential.

# Creating a database of all theorems, examples, exercises

- Can we take one lecture note latex source and extract in a structured way all theorems, proofs, exercises... [prototype done]
- Can we do it for all available online (probably yes, in progress)

# Expanding database - example

- Take the following proposition and proof. Organize the proof in logical steps:
- Take each of the steps and transform it into an exercise with a solution

# Proposition

Let $E$ and $F$ be vector spaces and let $T : E \to F$ be a linear map. The following sets are linear spaces:

1. $\{x \in E : T(x) = 0\}$ – this set is called the kernel of $T$, and is denoted by kernel $T$.
2. $\{y \in F : y = T(x), \text{ for some } x \in E\}$ – this set is called the image or range of $T$, and is denoted by image $T$.

**Exercise 1: Show that the kernel of a linear map is a subspace.**

Let $E$ and $F$ be vector spaces and let $T : E \to F$ be a linear map. Show that the kernel of $T$, denoted by kernel $T = \{x \in E : T(x) = 0\}$, is a subspace of $E$.

**Solution:**

To show that kernel $T$ is a subspace, it suffices to show that it is closed under the vector space operations:

- If $x_1, x_2 \in$ kernel $T$, then $T(x_1 + x_2) = T(x_1) + T(x_2) = 0$, by linearity. So, $x_1 + x_2 \in$ kernel $T$.
- For any scalar $\lambda$ and $x \in$ kernel $T$, we have $T(\lambda x) = \lambda T(x) = 0$. Thus, $\lambda x \in$ kernel $T$.

# Programming - thinking functionally

- The latex parser was written in Python and has around 1000 lines of code
- The code was almost 100% written in GPT4 with minimal human intervention from English descriptions.
- The key to write the code efficiently was to think functionally and write the code as a composition of functions
- Functional thinking reduces errors, allows for systematic testing and ease of change.

# Programming - Example

"Write a python code that takes a latex string and returns a list with all labels used in the latex string. Write also a unit test. Document all code with numerous comments."

```
import re

def extract_labels(latex_string):
    """
    Extracts all labels used in a LaTeX string.

    Args:
        latex_string (str): The LaTeX string to process.

    Returns:
        list: A list containing all the labels found in the LaTeX string.
    """
    # Regular expression pattern to match LaTeX label command
    pattern = r'\\label\{([^}]+)\}'

    # Find all occurrences of the pattern in the LaTeX string
    matches = re.findall(pattern, latex_string)

    # Return the list of labels
    return matches

# Unit Test
def test_extract_labels():
    latex_string = '''
    \begin{equation}
    E = mc^2 \label{mass_energy_equivalence}
    \end{equation}

    \begin{figure}
    \includegraphics{example.png}
    \caption{Example figure}
    \label{fig:example}
    \end{figure}

    As shown in Equation \ref{mass_energy_equivalence} and Figure
     \ref{fig:example}...
    '''

    expected_labels = ['mass_energy_equivalence', 'fig:example']
    extracted_labels = extract_labels(latex_string)

    assert extracted_labels == expected_labels, f"Expected
     {expected_labels}, but got {extracted_labels}"

    print("Unit test passed.")

# Run the unit test
test_extract_labels()
```

# Conclusions

- Mathematics programs can play a significant role in the AI revolution by providing basic skills training to practitioners and developing high-level approaches that address the core foundations.
- Various mathematical questions in AI are closely related to the research conducted in mathematics across different contexts.
- The exciting advancements in language models may have a considerable impact on mathematics, potentially at the level of theorem proving.
- The functional programming paradigm is expected to substantially accelerate research.
- Due to LLMs, the entry barrier to research in AI applications has been significantly lowered.
- Mathematicians should seize the opportunity to explore and contribute to the advancements in mathematics and AI.