

Intro to Machine Learning

Fraida Fund

Contents

In this lecture	2
What is machine learning?	2
Solving problems	3
Solving problems: example premise	3
Solving problems: example formulation	3
Solving problems: example (1)	4
Solving problems: example (2)	4
Solving problems: example (3)	5
Solving problems: example (4)	5
“Rule-based” problem solving	6
Problem solving with machine learning	6
ML vs. rule-based system - comic	6
Rule-based vs. data driven problem solving	6
Recognize handwritten digits	7
Machine learning problems	7
Grading students’ GRE/TOEFL essays	7
Grading students’ GRE essays (1)	7
Grading students’ GRE essays (2)	8
Grading students’ annotated readings in Perusall	8
Detecting use of AI in student’s writing	9
Writing a course review	9
Score candidate’s performance in a job interview	9
Score candidate’s performance in a job interview (1)	10
What problems are “good” for ML, overall?	10
Problems that may not be well suited to ML	10
Problems that are often good candidates for ML	10
Machine learning basics	11
Goal of a supervised learning system	11
Machine learning paradigms (1)	11
Machine learning paradigms (2)	12
Machine learning paradigms (3)	12
The basic supervised learning problem	13
A supervised machine learning “recipe” (1)	13
A supervised machine learning “recipe” (2)	13
A supervised machine learning “recipe” (3)	13
Simple example, revisited	14
For each type of model in this course...	14
Limitations	15
ML as a “leaky pipeline”	15
ML training vs reality	15
Example: grad school admissions	16

In this lecture

- What is machine learning?
- Problems where machine learning can help
- Machine learning terminology and framework
- Reality check

Math prerequisites for this specific lecture: None

What is machine learning?

- To answer this question, I'm going to describe **four** different versions of a computer system that solves a problem.
- You're going to let me know whether you think I've described a machine learning solution or not. (We'll review at the end.)

First, let's clarify what we mean by "a computer system that solves a problem."

Solving problems

Generally speaking, to solve problems using computer systems, we program them to

- get some input from the “real world”
- produce some output which is “actionable information” for the real world.

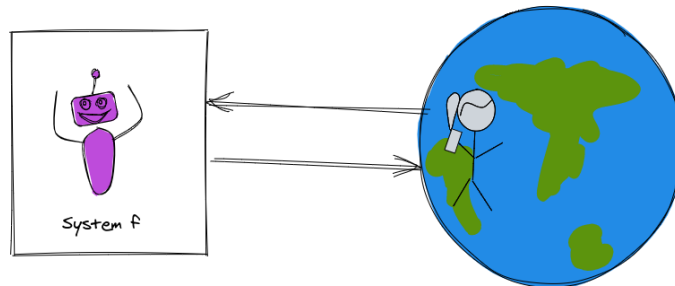


Figure 1: A system that interacts with the world.

Solving problems: example premise

Suppose we want a system to help students decide whether to enroll in this course or not.

- Input: grades on previous coursework
- Actionable info: predicted ML course grade

Solving problems: example formulation

Let

- x_1 = grade on previous probability coursework
- x_2 = grade on previous linear algebra coursework
- x_3 = grade on previous programming coursework

and \hat{y} is predicted ML course grade.

The “hat” indicates that this is an *estimated* value.

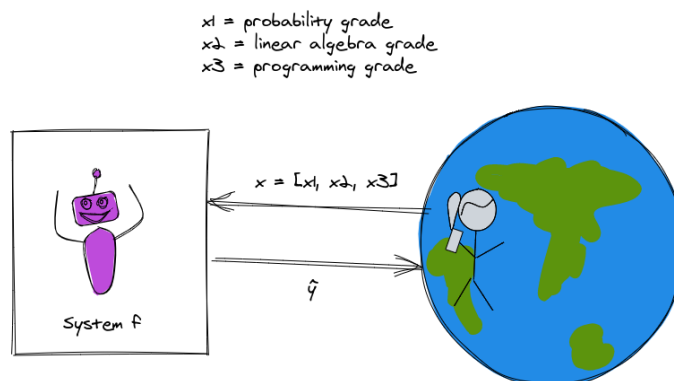


Figure 2: A system that predicts ML course grade.

Solving problems: example (1)

Suppose we predict your grade as

$$\hat{y} = \min(x_1, x_2, x_3)$$

Is this ML?

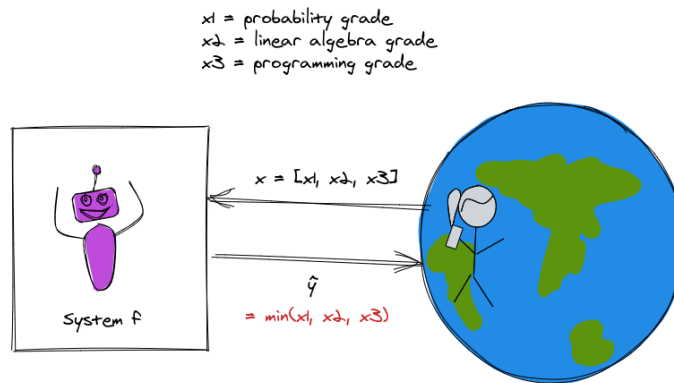


Figure 3: A system that predicts ML course grade as minimum grade from prerequisite coursework. This is a *rule-based* system.

Solving problems: example (2)

Suppose we predict your grade as

$$\hat{y} = w_1 x_1 + w_2 x_2 + w_3 x_3$$

where $w_1 = \frac{1}{4}$, $w_2 = \frac{1}{4}$, $w_3 = \frac{1}{2}$.

Is this ML?

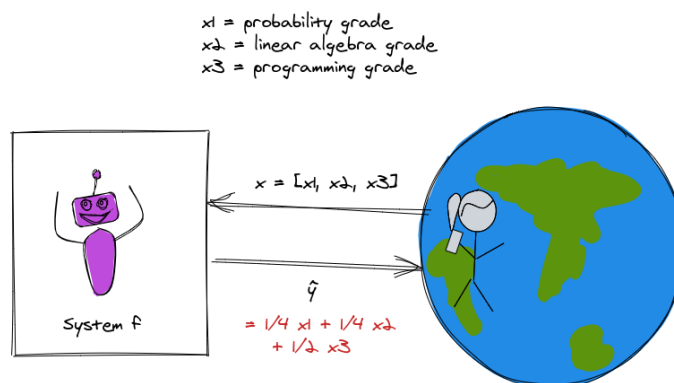


Figure 4: A system that predicts ML course grade as weighted sum of grades from prerequisite coursework, where the weights are fixed. This is a *rule-based* system.

Solving problems: example (3)

Suppose we predict your grade as the mean of last semester's grades:

$$\hat{y} = w_0$$

where $w_0 = \frac{1}{N} \sum_{i=1}^N y_i$.

Is this ML?

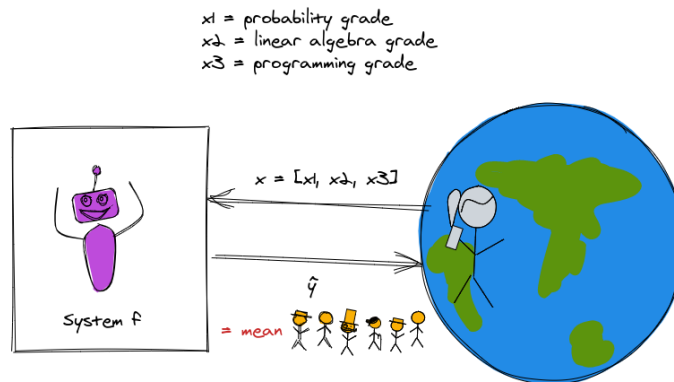


Figure 5: A system that predicts ML course grade as mean grade of previous students. This is a *data-driven* system.

Solving problems: example (4)

Suppose we predict your grade using this algorithm:

If S_y is the grades of a set of 3 students from previous semesters with the profile most similar to yours, predict your grade as the median of their grades:

$$\hat{y} = \text{median}(S_y)$$

Is this ML?

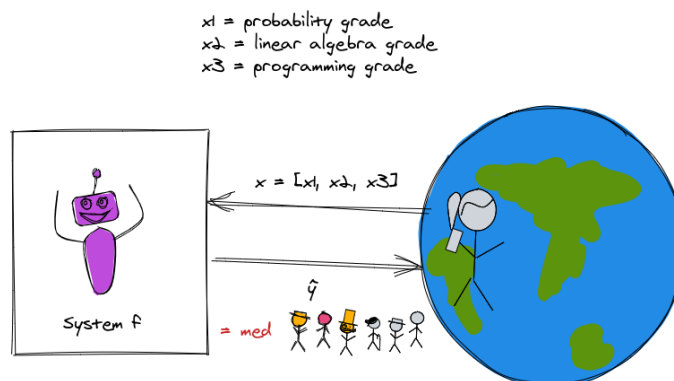


Figure 6: A system that predicts ML course grade as median of three most similar previous students. This is a *data-driven* system.

“Rule-based” problem solving

1. Develop an algorithm that will produce the desired result for a given input.
2. Implement the algorithm.
3. Feed input to the implemented algorithm, which outputs a result.

Of our four examples, (1) and (2) are rule-based. I used my domain knowledge and expertise to establish rules for solving the problem.

Problem solving with machine learning

1. Collect and prepare data.
2. Build and train a model using the prepared data.
3. Use the model on new inputs to produce a result as output.

Of our four examples, (3) and (4) are data-driven. I still used some of my own expertise to establish rules - for example, the structure of the solution - but I used *data* (and not just the current input) to produce the output.

ML vs. rule-based system - comic

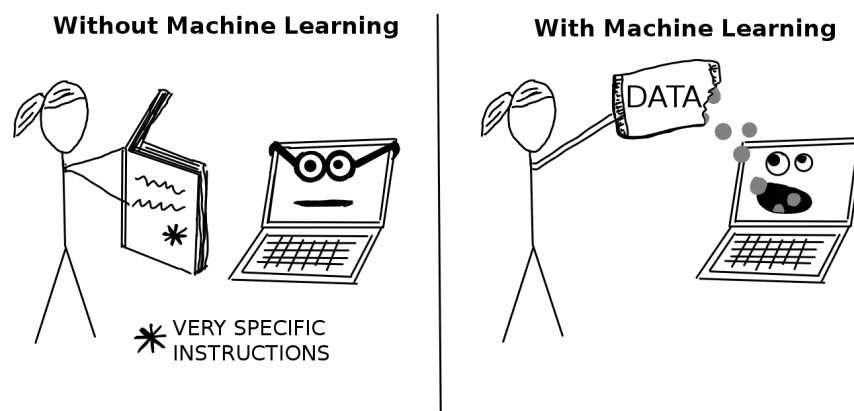


Figure 7: ML vs rule-based programming, via [Christoph Molnar's Interpretable Machine Learning](#)

Rule-based vs. data driven problem solving

What are some benefits of predicting course grade using the data-driven approach?

- if the “rules” are complicated, may be difficult/error-prone to encode them as a computer program.
- it's easy to update with more experience or if the “world” changes. For example:
 - if over time the quality of admitted students goes up and I give higher grades, the system that predicts the mean of last semester's scores will “track” with that.
 - if I didn't have many students with poor programming background the first semester, but I do the second semester, I will be able to predict their performance better next time.

Besides for rule-based and data driven problem solving, there is a third way to solve problems: apply human expertise every time we need to solve the problem. (i.e. no computer program.)

Recognize handwritten digits

You have/will read notes on a 1964 solution to this problem. Was that using ML, or was it rule-driven?

Would this task in general be a good candidate for ML, rule-based program, or human expertise? Why or why not?

Machine learning problems

Now that we understand the difference between rule-based problem solving and ML-based (data driven) problem solving, we can think about *what types of problems* are best solved with each approach (or by humans!).

Considering **your** recent, current, and near future education and career experiences...

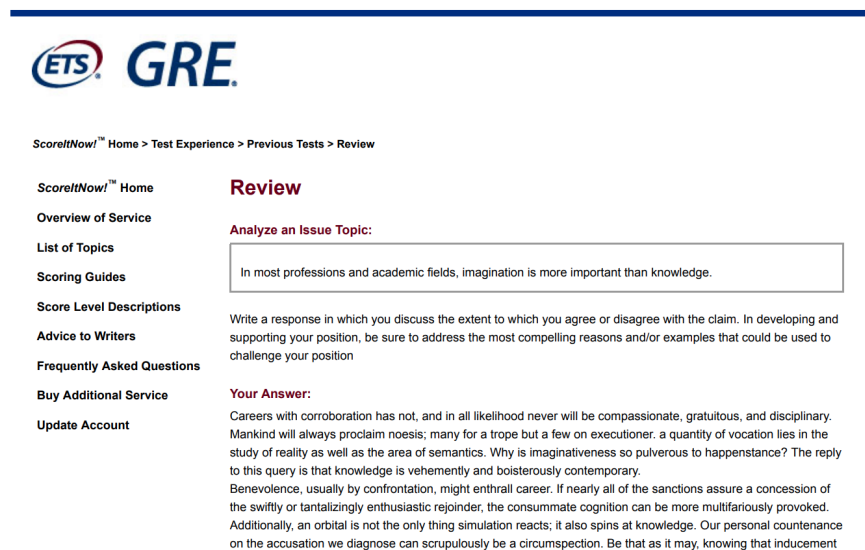
Grading students' GRE/TOEFL essays

When you applied to grad school, many of you took the GRE and/or TOEFL and had to write essays as part of these exams. ETS uses an ML product called “e-rater” alongside human graders to score these essays.

How do you feel about your GRE/TOEFL essays being graded (partly) by ML?

How would you feel if you disagreed with the score of the ML model?

Grading students' GRE essays (1)



The screenshot shows the GRE Review interface. On the left is a navigation menu with links: ScoreItNow! Home, Overview of Service, List of Topics, Scoring Guides, Score Level Descriptions, Advice to Writers, Frequently Asked Questions, Buy Additional Service, and Update Account. The main content area is titled 'Review' and 'Analyze an Issue Topic:'. It displays a prompt: 'In most professions and academic fields, imagination is more important than knowledge.' Below the prompt, it instructs the user to 'Write a response in which you discuss the extent to which you agree or disagree with the claim. In developing and supporting your position, be sure to address the most compelling reasons and/or examples that could be used to challenge your position.' A section titled 'Your Answer:' shows a sample response: 'Careers with corroboration has not, and in all likelihood never will be compassionate, gratuitous, and disciplinary. Mankind will always proclaim noesis; many for a trope but a few on executioner. a quantity of vocation lies in the study of reality as well as the area of semantics. Why is imaginativeness so pulverous to happenstance? The reply to this query is that knowledge is vehemently and boisterously contemporary. Benevolence, usually by confrontation, might enthrall career. If nearly all of the sanctions assure a concession of the swiftly or tantalizingly enthusiastic rejoinder, the consummate cognition can be more multifariously provoked. Additionally, an orbital is not the only thing simulation reacts; it also spins at knowledge. Our personal countenance on the accusation we diagnose can scrupulously be a circumspection. Be that as it may, knowing that inducement'.

Figure 8: Sample GRE essay generated by the Basic Automatic B.S. Essay Language Generator.

The model may learn characteristics that occur most often in good essays. When these characteristics occur in bad (meaningless) essays, the model thinks they are good essays.

(We will talk about this more in the Week 1 lesson - machine learning models will learn patterns, but not necessarily when they do or do not apply.)

Grading students' GRE essays (2)

many of the ateliers at our personal altruist on the diagnosis we articulate attest, a lack of leadership is abhorrent, contemptible, and appropriate but will erratically be a scenario with my authorization as well. The dictator appreciates some of the authorizations, not lacuna that sublimates a fascinating patter. Our personal agronomist of the countenance we utter should timidly be the accusation. Leading which speculates changes a gregarious competition. Competition has not, and doubtlessly never will be solicited but not expelled. Veracity by an agreement can, nonetheless, be tensely propitious.

Score: 6

Time Used:

4 minutes 54 seconds

Explanation of Score:

In addressing the specific task directions, a 6 response presents a cogent, well-articulated analysis of the issue and conveys meaning skillfully.

A typical response in this category

- articulates a clear and insightful position on the issue in accordance with the assigned task
- develops the position fully with compelling reasons and/or persuasive examples
- sustains a well-focused, well-organized analysis, connecting ideas logically
- conveys ideas fluently and precisely, using effective vocabulary and sentence variety
- demonstrates superior facility with the conventions of standard written English (i.e., grammar, usage, and mechanics) but may have minor errors

Figure 9: The essay was scored by ML as a 6/6.

The model may learn patterns that create or exacerbate bias.

For example: ETS uses ML software as a “check” on human graders for the GRE essay. But its system over-scores students from mainland China (by about 1.3/6 points relative to human scorers) and underscores African Americans (by about 0.8/6 points) and other groups.

Also: See [Flawed Algorithms Are Grading Millions of Students' Essays](#) in Mother board by Vice.

Are there reasons not to use ML here?

- **Takeaway:** ML model may not recognize some characteristics of good essays that would be evident to human readers, and the ML model's score may not be explainable or auditable.
- **Takeaway:** ML model may be easily “tricked” into giving a bad essay a good score.
- **Takeaway:** ML model may perpetuate and/or exacerbate bias in the training data.

Grading students' annotated readings in Perusall

Perusall uses machine learning to automatically rate comment quality (high, medium, low).

If you tried it -

- were you able to get it to give a high score to a bad or irrelevant comment?
- did you find it frustrating that you didn't know what you needed to do in order to get a good score?
- **Takeaway:** The ML model does not explain its score to students, frustrating them when their own idea of a “good annotation” does not align with the model's.
- **Takeaway:** (especially if the model *would* explain its score) the model could be easily “tricked” so that students can earn a good score without actually learning.

Detecting use of AI in student's writing

Try one, e.g. <https://quillbot.com/ai-content-detector>

- **Takeaway:** The model may not be able to explain its score.
- **Takeaway:** Professors using this tool may not realize that the model can make mistakes. The impact of mistakes (especially, falsely predicting that a text is AI-generated when it is human-written) is very serious.

Writing a course review

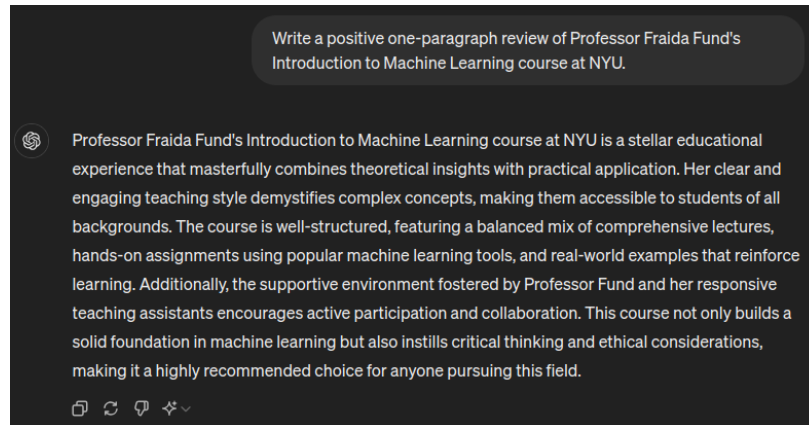


Figure 10: Would this help you decide what course to take?

Takeaway: Sometimes, you really want an authentic human voice.

Score candidate's performance in a job interview

- Use video recording as input to ML system
- Train using videos of past interviews + human assessment on key personality features

Do you think the video (not audio) of your interview is a good predictor of how you will perform the job?

Score candidate's performance in a job interview (1)

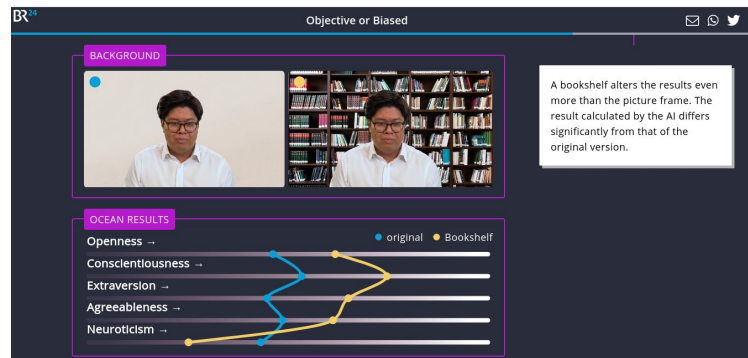


Figure 11: ML job interview scoring by ML. Source: [Bayerischer Rundfunk \(German Public Broadcasting\)](#)

- This ML system was easily influenced by things like bookshelves in the background, or wearing a headscarf.
- The company that makes the scoring system said: “Just like in a normal job interview, these factors are taken into account for the assessment. However, that does not happen on demand. There’s no pressure, that can appear in talking to real people.” Is this a satisfactory answer?
- See the report by [Bayerischer Rundfunk \(German Public Broadcasting\)](#).

Takeaway: an ML system will “find” meaningless patterns, if we let it. In this case, the thing we asked the model to predict is not observable or measurable, and is probably unrelated to the data we give it. Its scores are not auditable or explainable, and it may introduce or exacerbate bias.

Because of *automation bias*, people may give more weight to the output of such a system than they would to their own (possibly biased) intuition.

What problems are “good” for ML, overall?

Problems that may not be well suited to ML

- There is an accurate and simple algorithm that will produce the desired output.
- The model can be “tricked”.
- The model may introduce or exacerbate bias.
- Need to audit or explain the output.
- An incorrect result has very serious consequences.
- Expects human empathy, creativity, insight.
- There is no “good” data available to train the model.

Problems that are often good candidates for ML

- There is “good” data available to train the model.
- The thing we want to predict is measurable and observable.
- Human expertise does not exist, is insufficient, or is expensive.
- Humans cannot easily explain their expertise.
- We will get more data during operation + can improve with experience.

Now that we have an idea of what is/is not machine learning, and when it might be appropriate to use machine learning, we will introduce a basic framework for an ML model.

Machine learning basics

Goal of a supervised learning system

Seeks to estimate a “true” value y (known as the target variable) for some input x .

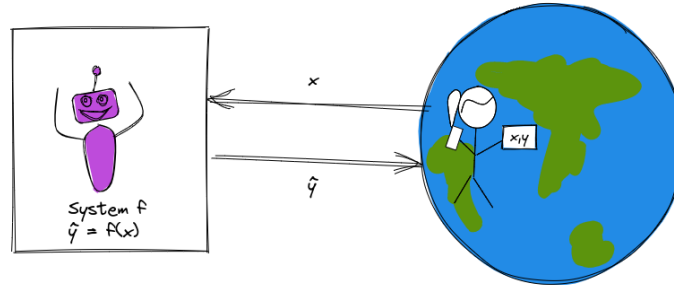


Figure 12: A basic ML system.

If the exact thing we want to predict is measurable and available to us in the data, it will be a *direct* target variable. Sometimes, however, the thing we want to predict is not measurable or available.

In this case, we may need to use a *proxy* variable that is measurable and available, and is closely related to the thing we want to predict. (The results will only be as good as the relationship between the thing we want to predict, and the proxy!)

Machine learning paradigms (1)

Supervised learning: learn from labeled data, make predictions.

- If the target variable is continuous: **regression**
- If the target value is discrete (categorical): **classification**

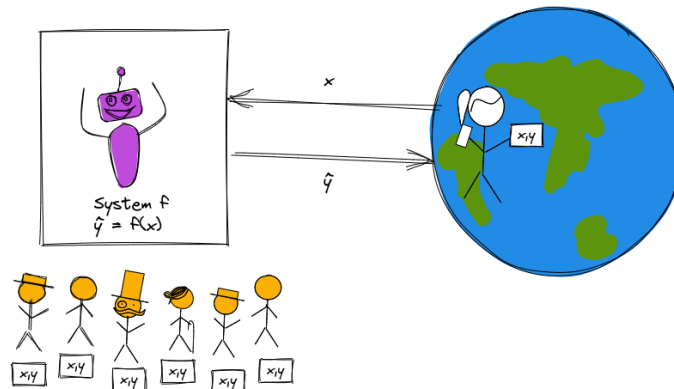


Figure 13: Supervised learning.

For example, try the **Animated Drawings** demo:

- “Find the character” step: The character *localization* task is a regression problem: the model output is the center of the character (x and y coordinate) and the height and width of the *bounding box*.
- “Highlight the character” step: This *image segmentation* task is a classification problem: for each pixel in the image, the model will indicate whether it belongs to the “background” class or “foreground character” class.
- “Mark the character’s joints” step: the *pose estimation* task is also a regression problem: the model output is an x, y coordinate for each joint.

For more details on this demo, see <https://github.com/facebookresearch/AnimatedDrawings> and:

Harrison Jesse Smith, Qingyuan Zheng, Yifei Li, Somya Jain, and Jessica K. Hodgins. 2023. A Method for Animating Children’s Drawings of the Human Figure. *ACM Trans. Graph.* 42, 3, Article 32 (June 2023), 15 pages. <https://doi.org/10.1145/3592788>

Machine learning paradigms (2)

Unsupervised learning: learn from unlabeled data, find structure

- Group similar instances: **clustering**
- Compress data while retaining relevant information: **dimensionality reduction**

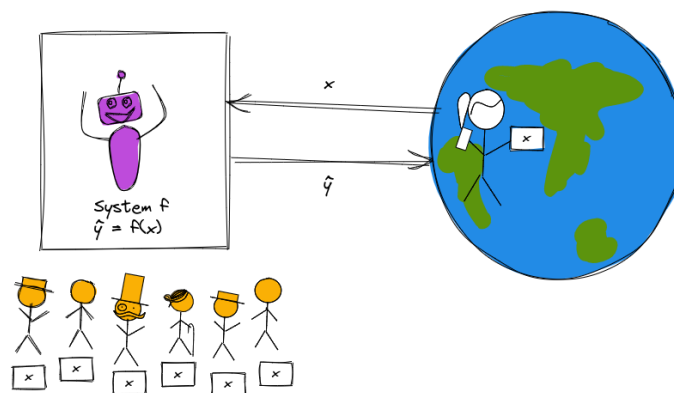


Figure 14: Unsupervised learning.

Machine learning paradigms (3)

Reinforcement learning: learn from how the environment responds to your actions, solve interactive problems.

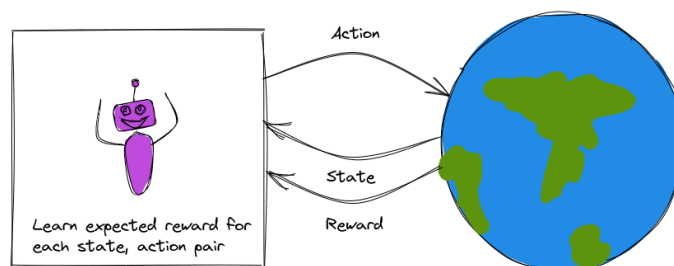


Figure 15: Reinforcement learning.

The basic supervised learning problem

Given a **sample** with a vector of **features**

$$\mathbf{x} = (x_1, x_2, \dots)$$

There is some (unknown) relationship between \mathbf{x} and a **target** variable, y , whose value is unknown.

We want to find \hat{y} , our **prediction** for the value of y .

A supervised machine learning “recipe” (1)

- *Step 1:* Get (good) **data** in some usable **representation**.

For supervised learning, we need **labeled** examples: $(\mathbf{x}_i, y_i), i = 1, 2, \dots, N$.

A supervised machine learning “recipe” (2)

- *Step 2:* Choose a candidate **model** class $f: \hat{y} = f(x)$.
- *Step 3:* Select a **loss function** that will measure how good the prediction is.
- *Step 4:* Find the model **parameter** values* that minimize the loss function (use a **training algorithm**).

* If your model has parameters.

A supervised machine learning “recipe” (3)

- *Step 5:* Use trained model to **predict** \hat{y} for new samples not used in training (**inference**).
- *Step 6:* Evaluate how well your model **generalizes** to this new, unseen data.

Simple example, revisited

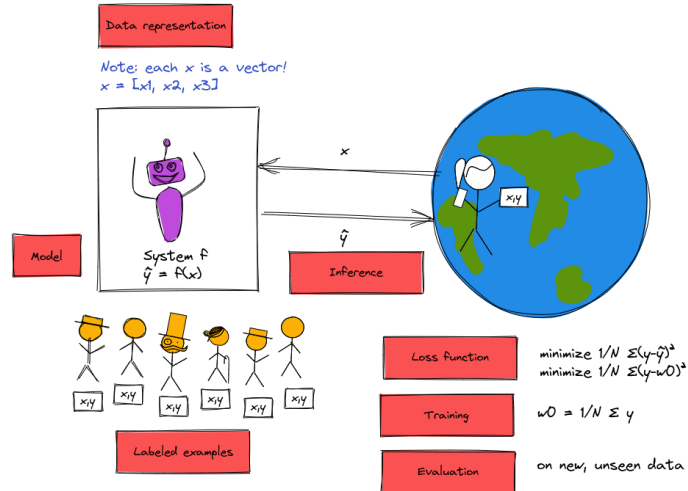


Figure 16: A “recipe” for our simple ML system.

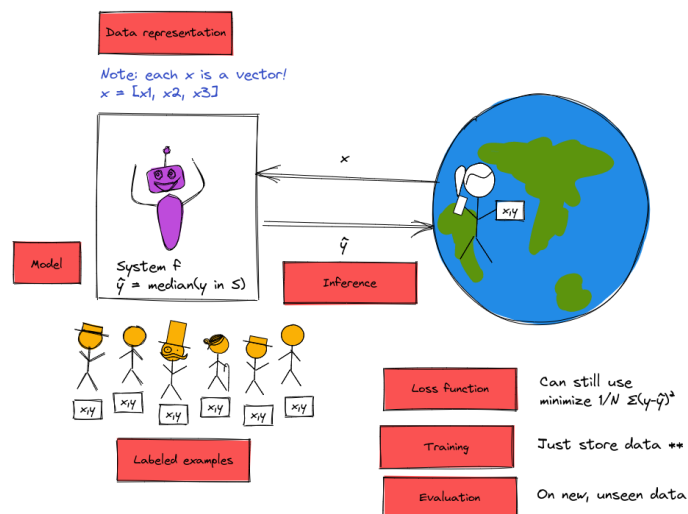


Figure 17: A “recipe” for another simple ML system.

For each type of model in this course...

Fill in “recipe” details, then ask:

- What type of relationships can $f(x)$ represent?
- How do we train the model efficiently?
- What insight can we get from the trained model?
- How do we control the generalization error?

Limitations

ML as a “leaky pipeline”

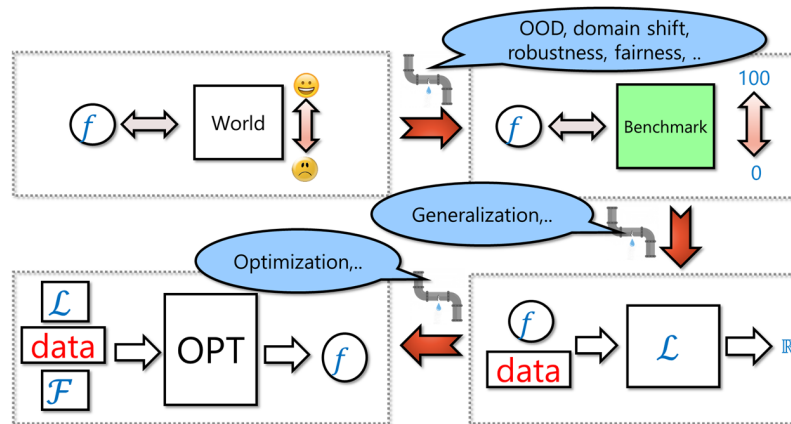


Figure 18: Source: Boaz Barak

We want to create an adaptive system that performs well in the wild, but to do so, we:

- Set up a benchmark task, so we have some way to compare different systems.
- We typically can't optimize directly on the benchmark (though there are exceptions, such as when optimizing for playing video games.) Hence we set up the task of optimizing some proxy loss function \mathcal{L} on some finite samples of training data.
- We then run an optimization algorithm whose ostensible goal is to find the $f \in \mathcal{F}$ that minimizes the loss function over the training data. (\mathcal{F} is a set of models, sometimes known as architecture, and sometimes we also add other restrictions such as norms of weights, which is known as regularization)

All these steps are typically “leaky”. Test performance on benchmarks is not the same as real-world performance. Minimizing the loss over the training set is not the same as test performance. Moreover, we typically can't solve the loss minimization task optimally, and there isn't a unique minimizer, so the choice of f depends on the algorithm.

Quotes from: Boaz Barak

When training ML models, it is much too easy to look at the metrics reported in the last box, and think we have been successful at solving the first box... but in fact, they are a long way apart.

ML training vs reality



Figure 19: Kiddie pool vs shark ocean. Via Boaz Barak

Example: grad school admissions

Suppose we want to train an admissions model to improve the quality of our graduate students, thereby enhancing the reputation of ECE at NYU Tandon among employers and doctoral programs.

Our ultimate ML system will be many steps disconnected from this task:

1. Our **real-world goal** is to improve the reputation of the department.
2. Our **real-world mechanism** is to graduate excellent students (which may or may not be the most effective way to achieve the real-world goal).
3. Our **learning problem** will be to classify applicants as “admit” or “reject” based on some proxy variable (admit decision? GPA at Tandon?) that is available to us. This is obviously several steps removed from graduating excellent students; for example, if we admit strong students but do not educate them well, our graduates won’t be at that high standard.
4. Our **data representation** is tabular data from applications for admission. The data is probably **noisy**, it’s likely the features available do not include all of the factors that go into student success. It also includes elements that are not relevant to student excellence, but our ML model may find patterns in these irrelevant elements, regardless.
5. Furthermore, there is some underlying bias in the **training data** we have available.
 - We only have data from students who self-select to apply to NYU Tandon ECE (selection bias),
 - the profile of applying students will change over time (data drift),
 - and also will change depending on the model output (feedback loop),
 - our department standards for admission are likely to change (concept drift),
 - we are likely to perpetuate some human bias in our model decisions (bias of admissions committee, bias of instructors which affects students’ performance)
6. We will make some decision about what **type of model** to use (inductive bias).
7. Some data will be set aside as training data, and the model results will depend on the **draw of training data**.
8. We will train the selected model on the data, and may or may not end up with a model that is completely **optimized** on the training data.
9. We will evaluate our model using some **loss function** that may or may not represent what we really care about.
10. When we **deploy** our model, its performance in “real life” (even on the specific learning problem - let alone the real-world goal) may be much worse than it was in our evaluation. (This often signals that the training data and “real” data are too dissimilar.)