

# Ling/CSE 472: Introduction to Computational Linguistics

---

April 14  
ML

# Overview

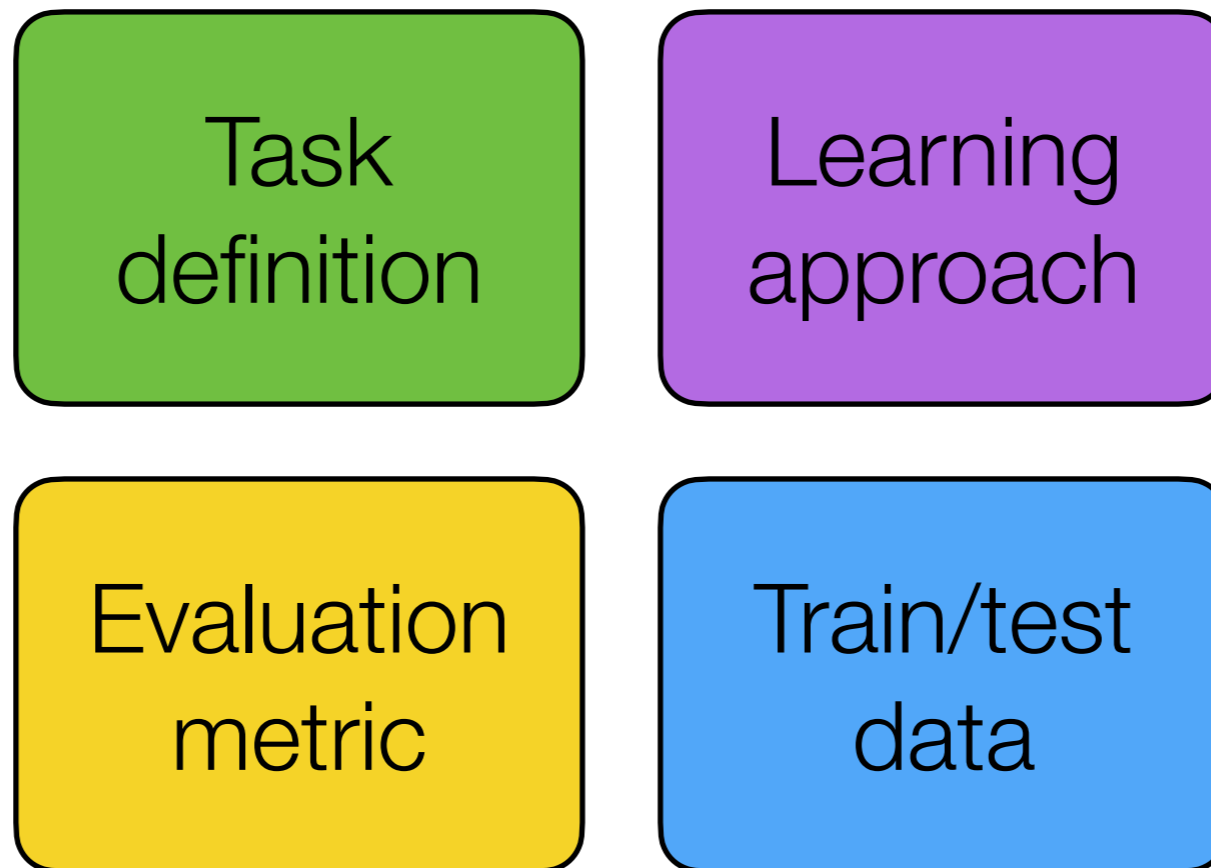
---

- ML
  - Big picture / then zooming out
  - A few key points
- reading questions
- NLP/compling in the news

# Machine learning, in a nutshell

---

- "Once the three components  $\langle T, P, E \rangle$  have been specified fully, the learning problem is well defined" (Mitchell, p.2)



# Machine learning, in context

---

Why do we care about this task?

How does dataset model the task?

-build something useful  
-learn about: computers, people, modeling domain

Task

Lear

Eval

Train

How does the task relate to the world?

How do we collect the data?

What happens when we deploy this?

# What is ML (Tom Mitchell)

---

- Study of algorithms that:
  - improve their performance
  - at some task
  - with experience
- Data → ML → **Understanding?**

# What is learning? (one definition)

---

- A process in which:
  - You observe
  - You extrapolate some knowledge from the observations
  - ...so that you can predict something about the future observations
- What is **machine** learning?

# What is machine learning?

---

- It is a process in which:
  - You represent observations as specific  $(x,y)$  pairs ( $x$  can be a vector of values)
  - You extrapolate some *mathematical function* from the observations
  - ...You can now feed to the function new data points, mapped to the same  $x$  representations, and it outputs  $y$
- But why? A mathematical equation is not real, is it? Why do we use it to represent the world? Why vectors, too?

# Presupposition check

---

## 1 Introduction

Machine learning is a discipline focused on two inter-related questions: “How can one construct computer systems that automatically improve through **experience?**” and “What are the fundamental theoretical laws that govern every learning system, **regardless of whether it is implemented in computers, humans or organizations?**” The study of machine learning is important both for addressing these fundamental scientific and engineering questions, and for the highly practical computer software it has produced and fielded across many applications.



# ML Tasks: Classification

---

- From data to discrete labels
  - Object detection
  - Weather prediction (e.g. rain, snow...)
  - For language tasks? (Poll on next slide)





# Tasks where we classify language (strings, documents)

W

Total Results: 0

Powered by  **Poll Everywhere**

Start the presentation to see live content. For screen share software, share the entire screen. Get help at [pollev.com/app](https://pollev.com/app)



# ML Tasks: Regression

---

- Predict a numeric value
  - Stock market
  - Weather prediction (temperature)
  - Airfares

# ML Tasks: Similarity

---

- Finding data
  - Given image, find similar ones
  - Similar products, songs...
  - Similar texts
  - Similar words...

# Machine Learning: Evaluation

---

- Human evaluation: How good is this output?
  - E.g. scores of fluency/adequacy for MT output
- Automatic evaluation: How well does this output match the stored gold standard?
  - Test data should be ‘held out’ (considered by neither the algorithm nor the developer)
    - Tests trained model’s ability to ‘generalize’
  - Metrics must be computable by comparing machine output to ground truth labels

# Train/dev/test splits

---

- “Test” is held out: don’t look, just report numbers
  - The parable of Sec 23 of the WSJ
- Train: training data, for training up the model
- Dev: development data
  - Parameter tuning
  - Error analysis
- Really small datasets: cross-validation

# Mitchell's (2017) key questions in ML

---

- How can computers improve performance through experience?
- Which theoretical laws govern learning systems?
- What are the key questions in NLP? Computational linguistics? Linguistics?

# ML: Key results

---

- No free lunch
  - “...no system has any basis to reliably classify new examples that go beyond those it has already seen...”
- Three sources of error: Bias, variance, and unavoidable error:
  - some probability of us being wrong
- Overfitting
  - When true error  $>$  train error
  - What is the relationship between true error and test error?



# Where is ML headed next?

---

- Will ML change the way we think about human learning?
- Human-machine (learning) interaction
- ML by reading
- Note that both directions involve natural language understanding

# Reading questions

---

- Can all of these different learning theories be used simultaneously, or are just one of them used at a time? I also assume each of the theories has certain strengths and weaknesses compared to each other?
- How are different types of machine learning models chosen for solving different problems, or what factors of a problem should be considered when choosing the strategy to approach it with machine learning techniques (supervised vs unsupervised/semisupervised, for example)? Some that I could think of are availability and type of training data, and maybe implementation constraints?

# Reading questions

---

- When does the ‘learning’ in machine learning happen? Is the algorithm ‘learning’ on the training data explicitly, or in scenarios where there is a user involved, for example, does the algorithm ‘learn’ through its interactions with the user?

# Reading questions

---

- What is the main point of having machine learning in computational linguistics, especially when it relates to something so complicated and at times as convoluted as human language? Is it for things like auto-fill to learn the words you most often use in succession? I also struggle to picture how it could relate to linguistics research and whether or not it has the possibility to learn things about a language that even we don't know since I would then assume that it shouldn't know either since we didn't teach it.

# Reading questions

---

- How much of the math and equations parts of these theories do I need to know about? Also is it possible to get some examples of how the most applicable theories for our purposes actually work? because I'm struggling to picture how these things work and their applications just going off of numbers, letters, and equations...

# Reading questions

---

- What exactly is a support vector machine classifier, and what are the ways that it uses machine learning to accomplish a task?
- I am struggling to visualize bayesian networks, and how this graph would look with "conditional independencies"?
- What exactly are gradient descent methods, and why does it need the help of a specialized gpu hardware?

# Reading questions

---

- The use of kernel methods to find non-linear relationships for regression or transformative models is a little confusing. As the kernel computes higher-dimension comparisons to evaluate similarity between data points, it seems better suited for classification tasks and I'm a little bit unsure on how it can be applied to other purposes (seq2seq for machine translation, regression, etc).
- Also, I'm a bit confused about why kernel methods work and what exactly they can be used for.
- One section in the reading that was the most confusing for me was the section about Bayesian Networks. Are those just generally a type of graphical model?
-

# Reading questions

---

- My understanding of semi-supervised learning is that it resides somewhere between supervised and unsupervised learning, where we have a model trained on supervised techniques with pre-labeled data but we want to augment its capabilities using additional data which is unlabeled. Is this correct and what are some real world applications where semi-supervised learning techniques can be used to improve the accuracy of an ML model?
- Is supervised learning always preferable to unsupervised learning? I'm wondering if a machine learning system for computational linguistics that was built with some knowledge of linguistics (the training data is annotated) would be better in some way than a machine learning system that was built without linguistics knowledge.



# Reading questions

---

- In the generative model part, we talked about Naive Bayes and Logistic Regression both "learns" the probability of  $Y$  given  $X$ . This is also seen in the behavior of chatGPT as it is like spitting out words one or more at a time. I am naively thinking that in the Large Language Model like that, the  $X$  needs to be extremely long to produce an accurate guess of  $Y$ . So, how long does that should be? How much does it take to train that? How environmentally costly is that?

# Reading questions

---

- I was most confused by the section on distant rewards and reinforcement learning. Beyond computation speed, what are the biggest factors in the speed of analyzing how good some chess move is? Are evaluation functions typically quite optimized? I would imagine for chess that the evaluation function is probably pretty optimized but I'm sure for other things there are unoptimized evaluation functions. Do evaluation functions have to look ahead at the possible states stemming from each currently possible state or would they only see what position you are in after a single move? For example, when evaluating some move in chess, does the evaluation function only look at that move or does it look at the moves stemming from that move and so on to some depth? I mainly focused on chess in this so it would be interesting to hear about some other applications for evaluation functions besides something like a board game.

# Reading questions

---

- For Q-learning, how is the  $Q(s, a)$  function created? For the example they give with the car on a slippery road, would  $Q(s, a)$  just be created from previous analyses of cars on slippery roads or is it made in some other way? Also, why wouldn't the learner be able to simulate a car on a slippery road? From my understanding of classical mechanics in physics, the future state of a car on a slippery road is deterministic if you know enough parameters in the system. Would the limitation in this case be that there are too many parameters to keep track of and use to calculate the future state?

# Reading questions

---

- The reading mentions under Key Results that no system, computer or human, can reliably classify new examples that go beyond the examples it's already seen in training. However, I've seen a lot of people all over the internet claiming the opposite. Where does this misunderstanding come from?
- Clarification on the difference between task T and experience E in the three components T P E.

# Reading questions

---

- The reading mentions "three sources of error in learned functions: bias, variance and unavoidable error" and I understand the basic idea but I'm struggling a little to visualize examples of the training data/"experience" which would lead to these sources of error, particularly the "unavoidable error" in the case of learning a non-deterministic function.

# NLP/compling in the news

---

- <https://www.ctinsider.com/politics/article/ct-chris-murphy-chatgpt-artificial-intelligence-17883105.php>
- <https://twitter.com/emilymbender/status/1644695247820296193>