Lecture 1.

Q: What is Machine Learning?

A: Engineering paradigm that replaces expert knowledge w/ prediction based on data.

e.g: OCR - Optical Character recognition.
- Less programming.
- Robust and easily adaptive
- Less dependent on expert knowledge.

e.g. め "me" in Japanese.

e.g. Face detection: Viola / Jones used abstract features fed into AdaBoost (abstract answer to theoretical question).

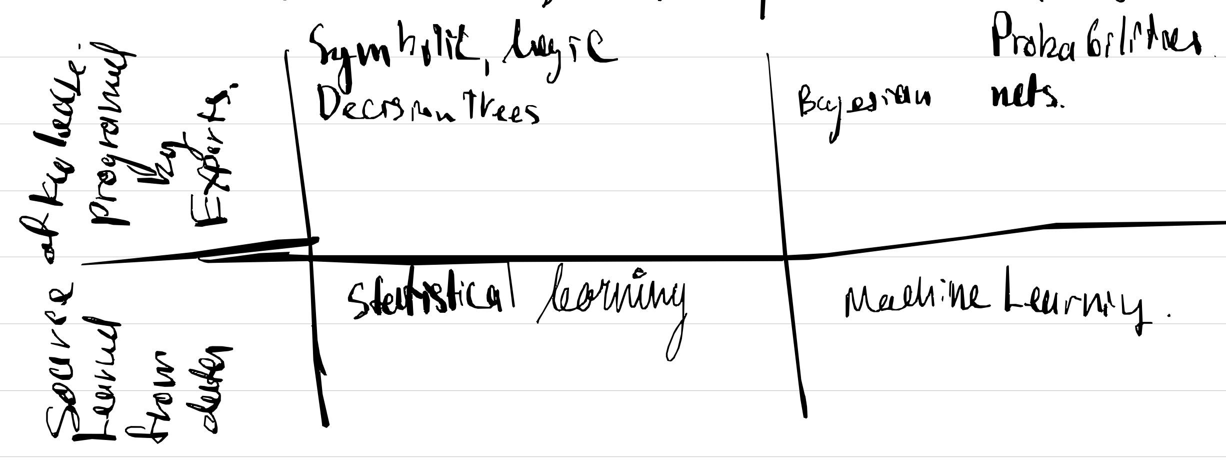
e.g. Machine Translation:
- Complex output.
- Learn from: - annotated translations
  - matching text
  - text (in a single language).

e.g. Depth from Stereo Images

A.I: Rule-based approach! (1950s - 2010s).
- eliza - first chat bot
- Dendral - Mass Spectrometry
- CYC - Machine Reasoning AI.

A.I: logic based vs probability.

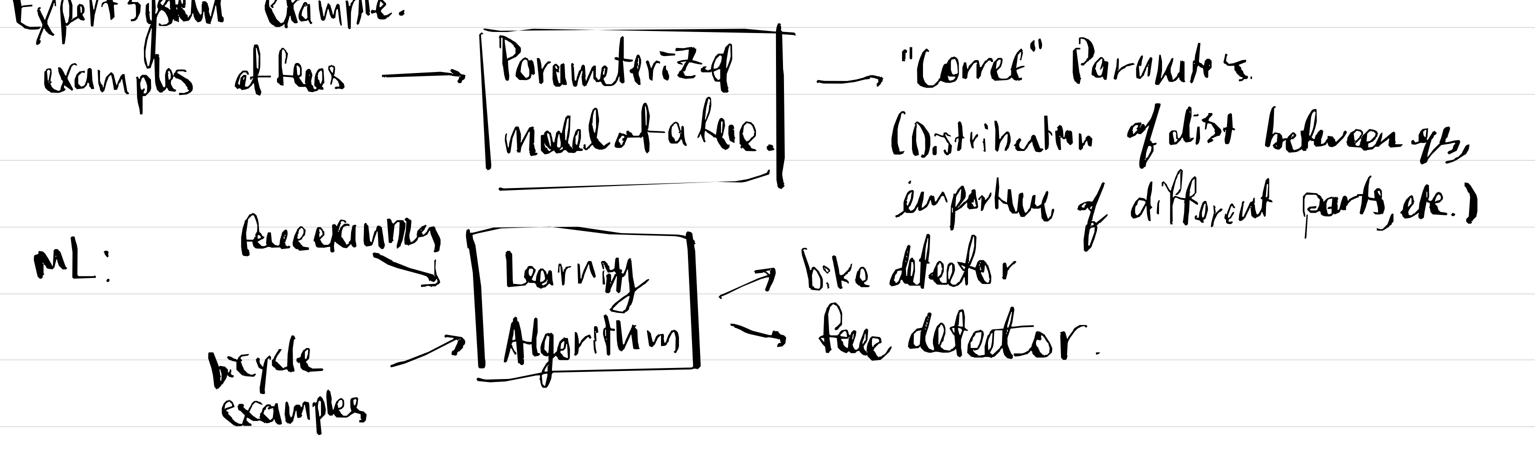
how knowledge is represented and used.



First Instance of "Machine Learning"
checkers algorithm.

Machine Learning: Less expert knowledge.

Expert system example:



Q: Can we develop a generic learning algorithm?

A: Two viewpoints!

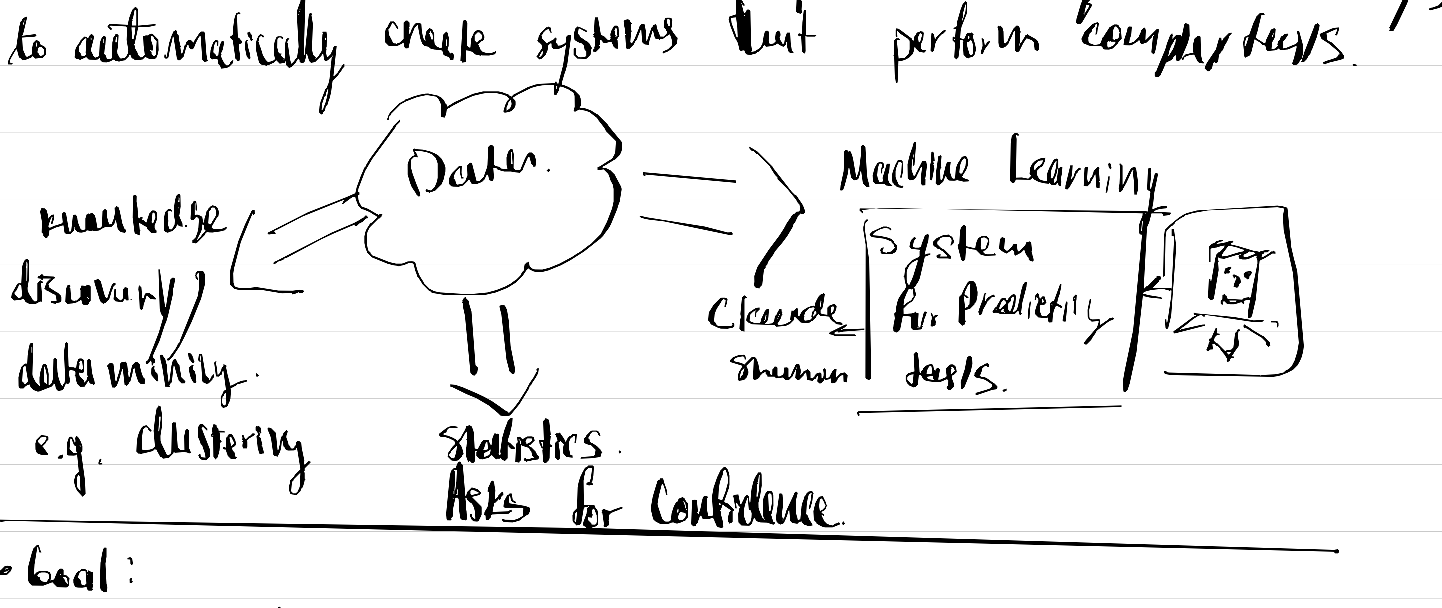
Chomsky: grammar is hard-wired

Hinton: ∃ some universal learning algorithm.

Who is right? Chomsky is closer. There is no free lunch. (theorem)
We need some expert knowledge.

Using less expert knowledge requires for greater compute resources.
Growth advances in compute (using GPU) has allowed Machine Learning.

"Machine Learning": Use data and examples, instead of expert knowledge, to automatically create systems that perform complex tasks.



- Goal:
  - understand concepts.
  - not tool box class.
- Theory:
  - Mathematical derivations
- Practice:
  - Half is weekly programming exercises
  - Other half is derivation!
- Recc Textbook:
  "Understanding Machine Learning", Shalev-Shwartz and Ben-David.
- Grades:
  - 20% Weekly Hw.
  - 20% Weekly online quizzes.
  - 50% Final exam
  - 10% Bonus
  - No midterm

Learning Predictors (Supervised Learning)
- Domain $\mathcal{X}$.
  - each $x \in \mathcal{X}$ is an instance.
  - e.g. set of all possible emails.
- Label Set $\mathcal{Y}$.
  - Focus on binary classification $\mathcal{Y} = \{\pm 1\}$
  - e.g. +1 means "SPAM" -1 means "Not SPAM"
- Predictor: $h: \mathcal{X} \to \mathcal{Y}$ i.e. $h \in \mathcal{Y}^{\mathcal{X}}$
  - "classifier" or "hypothesis"?
  - $h(x) = \begin{cases} +1 & \text{if contains "free"} \\ -1 & \text{else.} \end{cases}$

Online Learning Process.
- At each $t = 1, 2, \ldots$
  - Receive $x_t \in \mathcal{X}$ (receive email)
  - Predict a label $\hat{y}_t = h_t(x_t)$. (predict if it's spam)
  - We see correct label (User tells us if it was really spam)
  - Update predictor h_{t+1} based on (x_t, y_t).
- Learning rule: Mapping
  $A: (\mathcal{X} \times \mathcal{Y})^* \to \mathcal{Y}^{\mathcal{X}}$
  $h_t = A((x_1, y_1), \ldots, (x_{t-1}, y_{t-1}))$.
- Goal: Make few mistakes.
- Is this possible?
  $\mathcal{X} = \{\text{items in basket}\}$, $\mathcal{Y} = \{\text{Ian, Oliver}\}$.

	x_t	$\hat{y}_t$	y_t
(Yellow shirt)	1	I	O
(Orange hoodie)	2	I	O
(Yellow Buttondown)	3	O	I
(Jacket)	4	I	O

Turns out it's random! ∃ a predictor, but we cannot learn it.
We need more structure or knowledge.

No Free Lunch: Online Version.
- For any finite $\mathcal{X}$ w/ n ele. any learning rule A. ∃ a map over sequence on which A makes at least n mistakes.

Lecture 2.

No freelunch (Online Version):

- For any finite X w/ n ele. any learning rule A. ∃ a map over sequence on which A makes at least n mistakes.
- For any infinite X; for any learning rule ∃ a map and a sequence: A makes a mistake every time.

Prior knowledge:

- Assume $y_t = f(x_t)$ for some $f \in \mathcal{H}$.
- $\mathcal{H} \subseteq \mathcal{Y}^X$ is a hypothesis class.
  - Learner knows $\mathcal{H}$ but does not know f.
- $\mathcal{H}$ is our Expert knowledge.
- $\{(x_t, y_t)\}_t$ is realizable by $\mathcal{H}$ if $\exists f \in \mathcal{H} : \forall t \; y_t = f(x_t)$

e.g. $\mathcal{H}$ = all predictors based on single word occurrence.