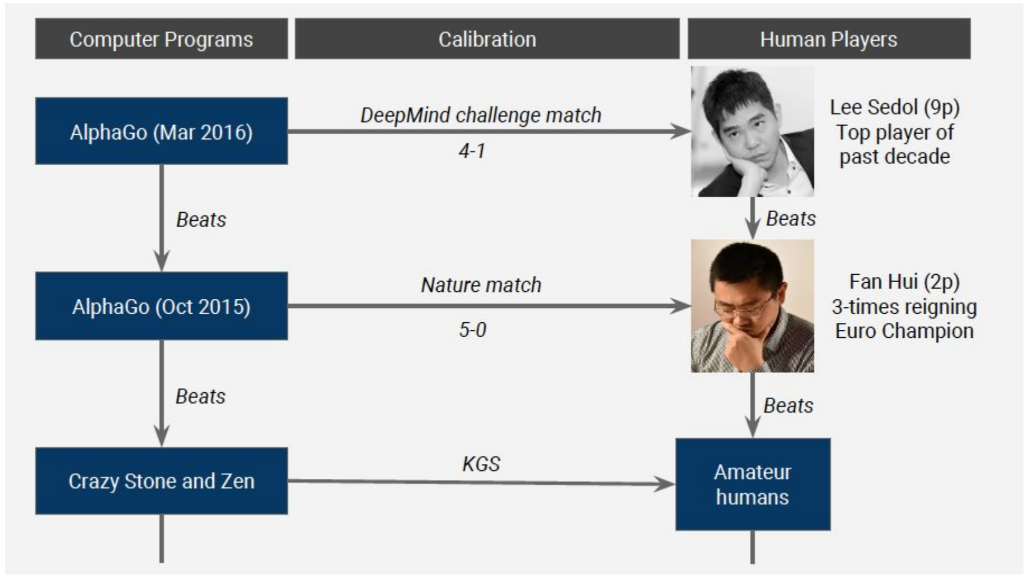
CS 188: Artificial Intelligence

Learning I:

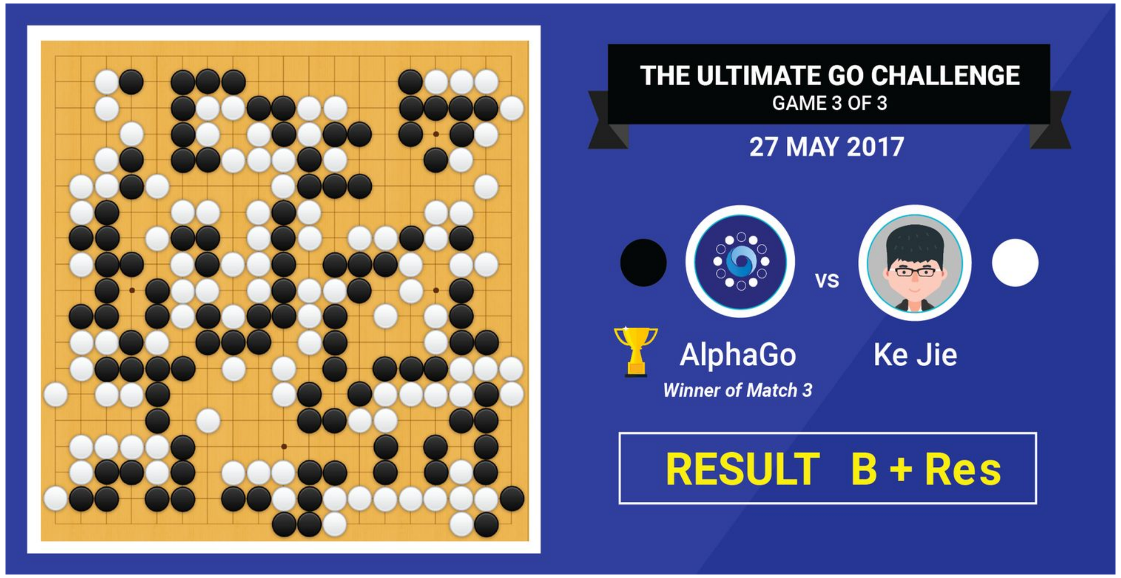
Instructors: Stuart Russell and Dawn Song

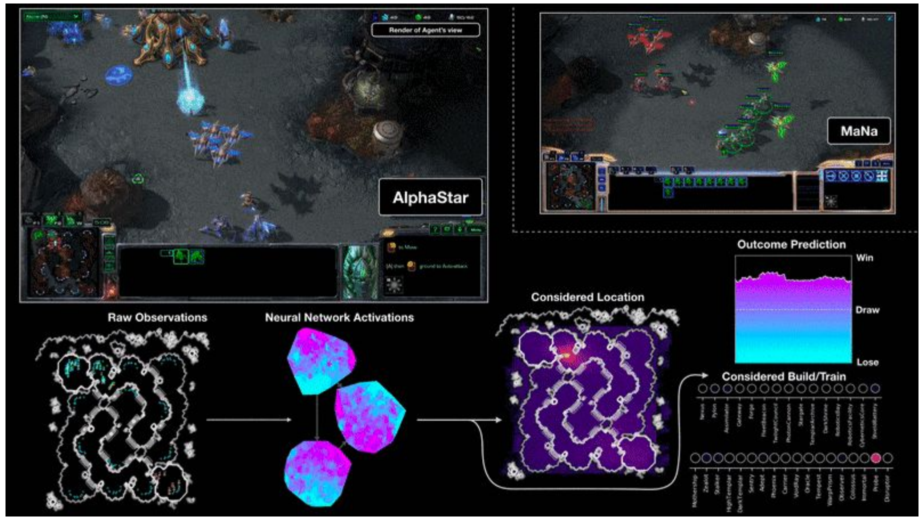
University of California, Berkeley

AlphaGo & AlphaStar: Winning over World Champions



Source: David Silver





Deep Learning Powering Everyday Products

pcmag.com

theverge.com

Lectures on learning

- ***Learning***: a process for improving the performance of an agent through experience
- Learning I (today):
  - The general idea: generalization from experience
  - Supervised learning: classification and regression
- Learning II: neural networks and deep learning
- Learning III: statistical learning and Bayes nets
- Reinforcement learning II: learning complex V and Q functions

Learning: Why?

- *The baby, assailed by eyes, ears, nose, skin, and entrails at once, feels it all as one great blooming, buzzing confusion …*

  [William James, 1890]

- Learning is ***essential*** in unknown environments – when the agent designer lacks omniscience

Learning: Why?

- *Instead of trying to produce a programme to simulate the adult mind, why not rather try to produce one which simulates the child's? If this were then subjected to an appropriate course of education one would obtain the adult brain. Presumably the child brain is something like a notebook as one buys it from the stationer's. Rather little mechanism, and lots of blank sheets.*

  [Alan Turing, 1950]

- Learning is *useful* as a system construction method, i.e., expose the system to reality rather than trying to write it down

Learning: How?

Mr. & Mrs., Skinner view daughter Debbie in a Skinner Box

Learning: How?

Key questions when building a learning agent

- What is the ***agent design*** that will implement the desired performance?
- Improve the performance of ***what piece*** of the agent system and ***how is that piece represented***?
- ***What data*** are available relevant to that piece? (In particular, do we know the right answers?)
- ***What knowledge*** is already available?

Examples

Agent design	**Component**	**Representation**	**Feedback**	**Knowledge**
Alpha-beta search	Evaluation function	Linear polynomial	Win/loss	Rules of game; Coefficient signs
Logical planning agent	Transition model (observable envt)	Successor-state axioms	Action outcomes	Available actions; Argument types
Utility-based patient monitor	Physiology/sensor model	Dynamic Bayesian network	Observation sequences	Gen physiology; Sensor design
Satellite image pixel classifier	Classifier (policy)	Markov random field	Partial labels	Coastline; Continuity scales

Supervised learning: correct answers for each training instance
Reinforcement learning: reward sequence, no correct answers
Unsupervised learning: "just make sense of the data"

Supervised learning

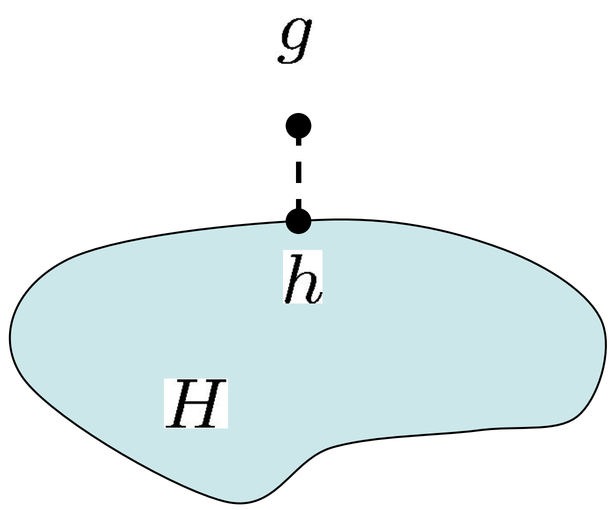
- To learn an unknown ***target function*** f
- Input: a ***training set*** of ***labeled examples*** (x_j,y_j) where $y_j = f(x_j)$
    - E.g., x_j is an image, $f(x_j)$ is the label "giraffe"
    - E.g., x_j is a seismic signal, $f(x_j)$ is the label "explosion"
- Output: ***hypothesis*** h that is "close" to f, i.e., predicts well on unseen examples ("***test set***")
- Many possible hypothesis families for h
    - Linear models, logistic regression, neural networks, decision trees, examples (nearest-neighbor), grammars, kernelized separators, etc etc
- ***Classification*** = learning f with discrete output value
- ***Regression*** = learning f with real-valued output value

Inductive Learning (Science)

- Simplest form: learn a function from examples
  - A target function: g
  - Examples: input-output pairs $(x, g(x))$
  - E.g. x is an email and $g(x)$ is spam / ham
  - E.g. x is a house and $g(x)$ is its selling price

- Problem:
  - Given a hypothesis space H
  - Given a training set of examples x_i
  - Find a hypothesis $h(x)$ such that $h \sim g$

- Includes:
  - Classification (outputs = class labels)
  - Regression (outputs = real numbers)

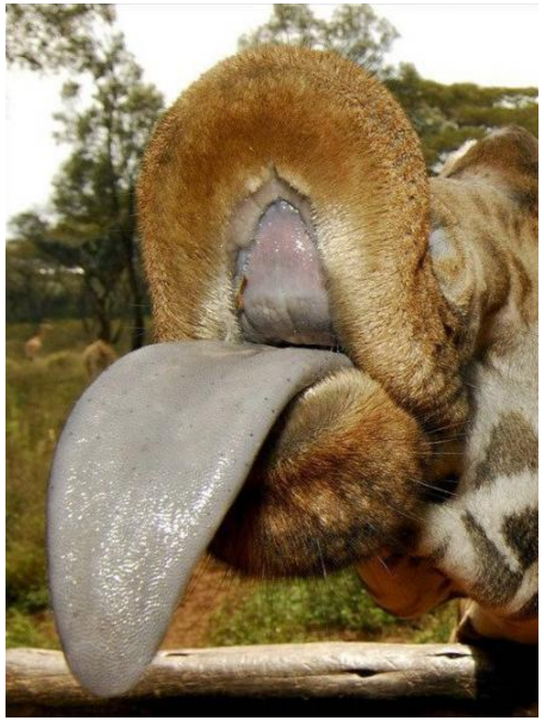


Classification example: Object recognition

x

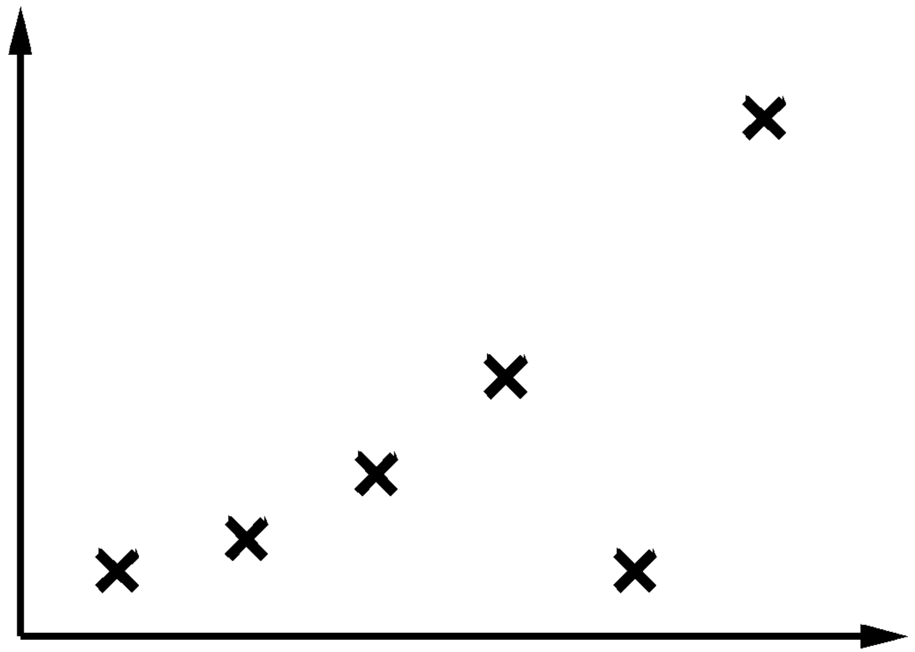
f(x)	giraffe	giraffe	giraffe	llama	llama	llama

X=

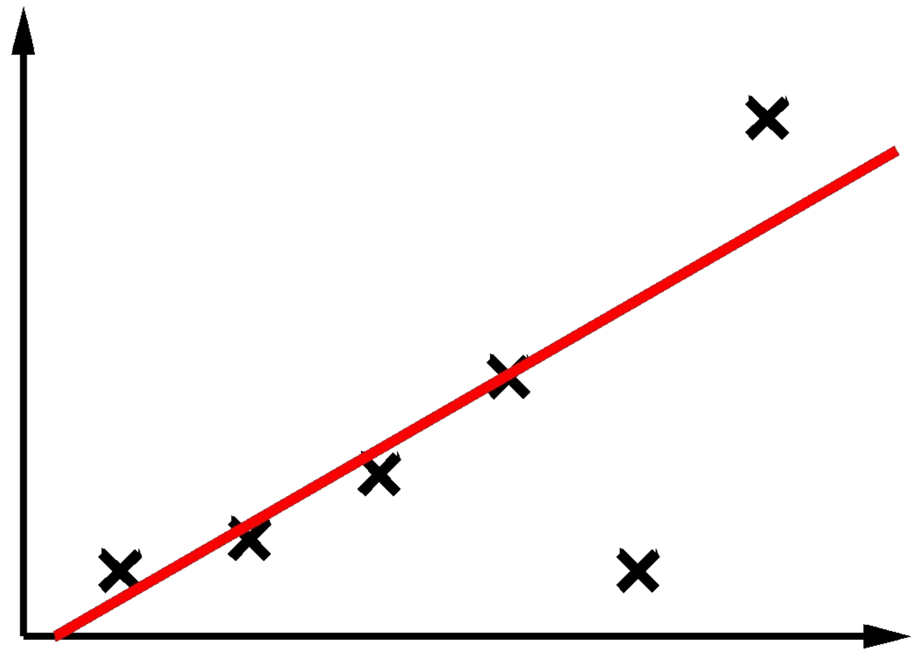


f(x)=?

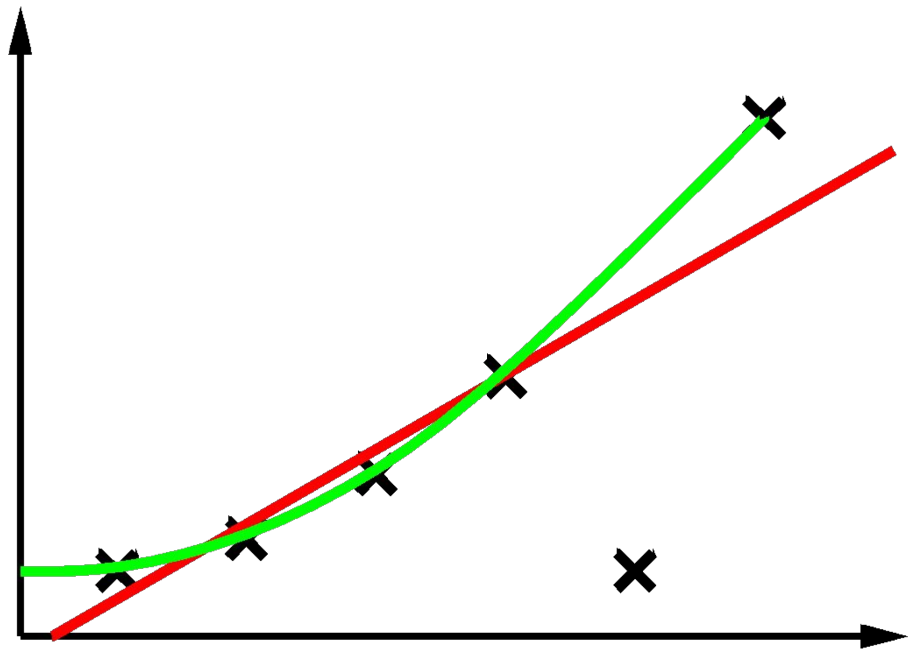
Regression example: Curve fitting



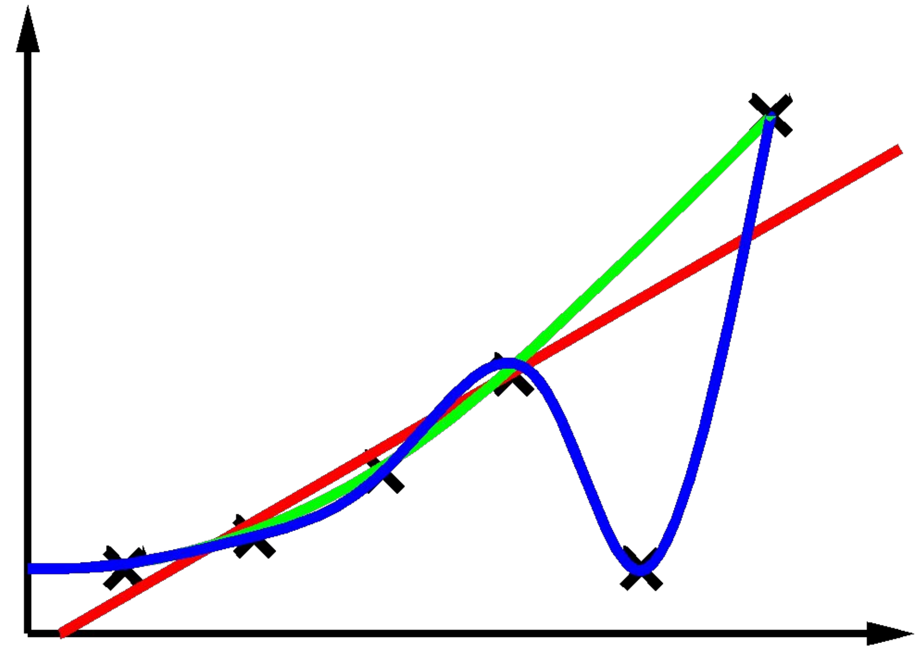
Regression example: Curve fitting



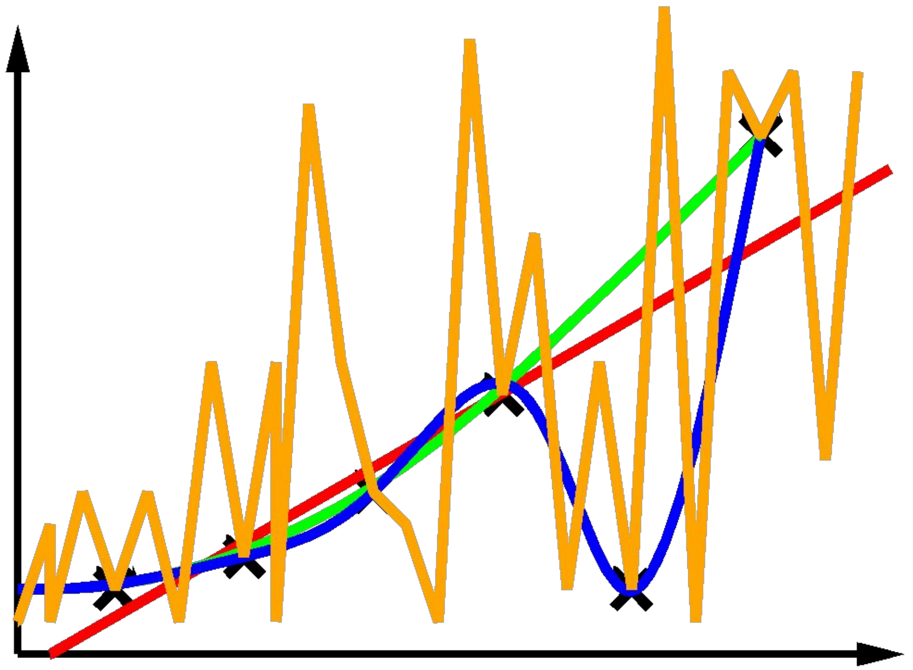
Regression example: Curve fitting



Regression example: Curve fitting



Regression example: Curve fitting



Consistency vs. Simplicity

- Fundamental tradeoff: bias vs. variance

- Usually algorithms prefer consistency by default (why?)

- Several ways to operationalize “simplicity”
  - Reduce the hypothesis space
    - Assume more: e.g. independence assumptions, as in naïve Bayes
    - Have fewer, better features / attributes: feature selection
    - Other structural limitations (decision lists vs trees)
  - Regularization
    - Smoothing: cautious use of small counts
    - Many other generalization parameters (pruning cutoffs today)
    - Hypothesis space stays big, but harder to get to the outskirts

Basic questions

- Which hypothesis space H to choose?
- How to measure degree of fit?
- How to trade off degree of fit vs. complexity?
  - "*Ockham's razor*"
- How do we find a good h?
- How do we know if a good h will predict well?

Classification

Example: Spam Filter

- Input: an email
- Output: spam/ham

- Setup:
  - Get a large collection of example emails, each labeled “spam” or “ham” (by hand)
  - Learn to predict labels of new incoming emails
  - Classifiers reject 200 billion spam emails per day

- Features: The attributes used to make the ham / spam decision
  - Words: FREE!
  - Text Patterns: $dd, CAPS
  - Non-text: SenderInContacts, AnchorLinkMismatch
  - …

✗

Dear Sir.

First, I must solicit your confidence in this transaction, this is by virture of its nature as being utterly confidencial and top secret. …

✗

TO BE REMOVED FROM FUTURE MAILINGS, SIMPLY REPLY TO THIS MESSAGE AND PUT "REMOVE" IN THE SUBJECT.

99 MILLION EMAIL ADDRESSES
FOR ONLY $99

✓

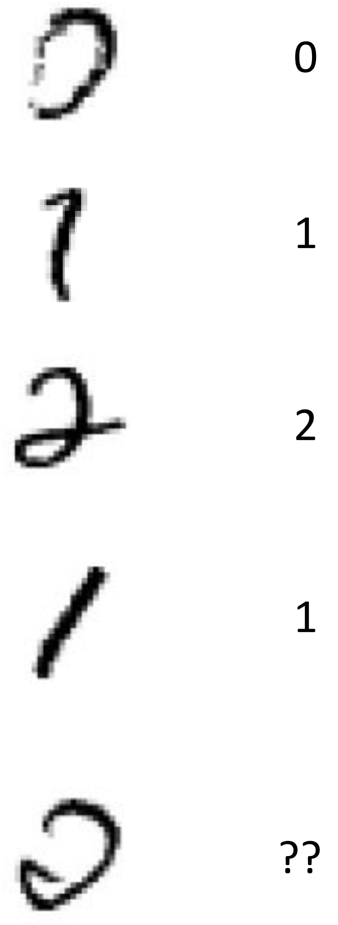
Ok, Iknow this is blatantly OT but I'm beginning to go insane. Had an old Dell Dimension XPS sitting in the corner and decided to put it to use, I know it was working pre being stuck in the corner, but when I plugged it in, hit the power nothing happened.

Example: Digit Recognition

- Input: images / pixel grids
- Output: a digit 0-9

- Setup:
  - MNIST data set of 60K collection hand-labeled images
    - Note: someone has to hand label all this data!
  - Want to learn to predict labels of new digit images

- Features: The attributes used to make the digit decision
  - Pixels: (6,8)=ON
  - Shape Patterns: NumComponents, AspectRatio, NumLoops
  - …



Other Classification Tasks

- Medical diagnosis
  - input: symptoms
  - output: disease
- Automatic essay grading
  - input: document
  - output: grades
- Fraud detection
  - input: account activity
  - output: fraud / no fraud
- Email routing
  - input: customer complaint email
  - output: which department needs to ignore this email
- Fruit and vegetable inspection
  - input: image (or gas analysis)
  - output: moldy or OK

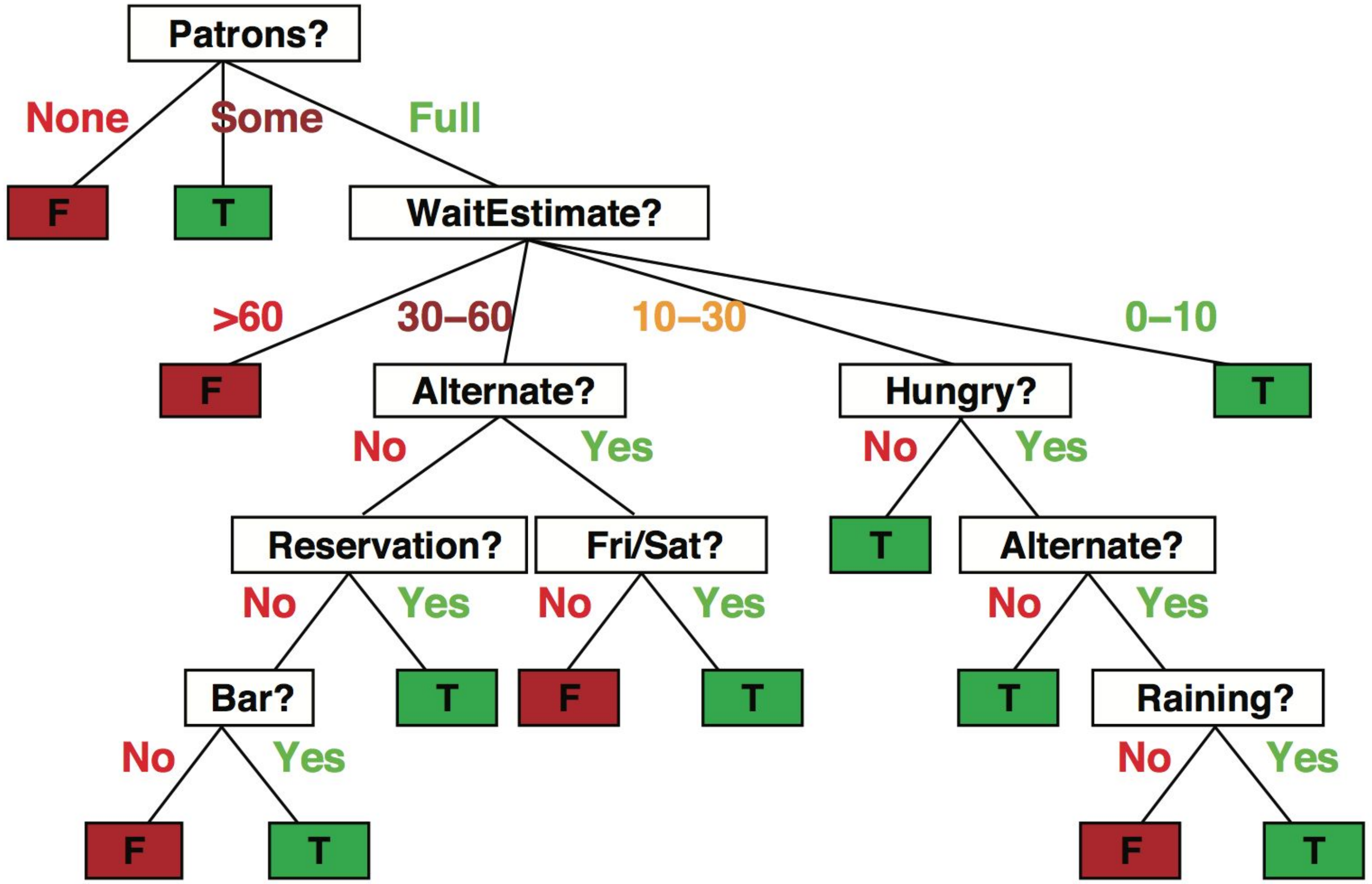
- … many more

Decision tree learning

- Decision tree models
- Tree construction
- Measuring learning performance

Decision trees

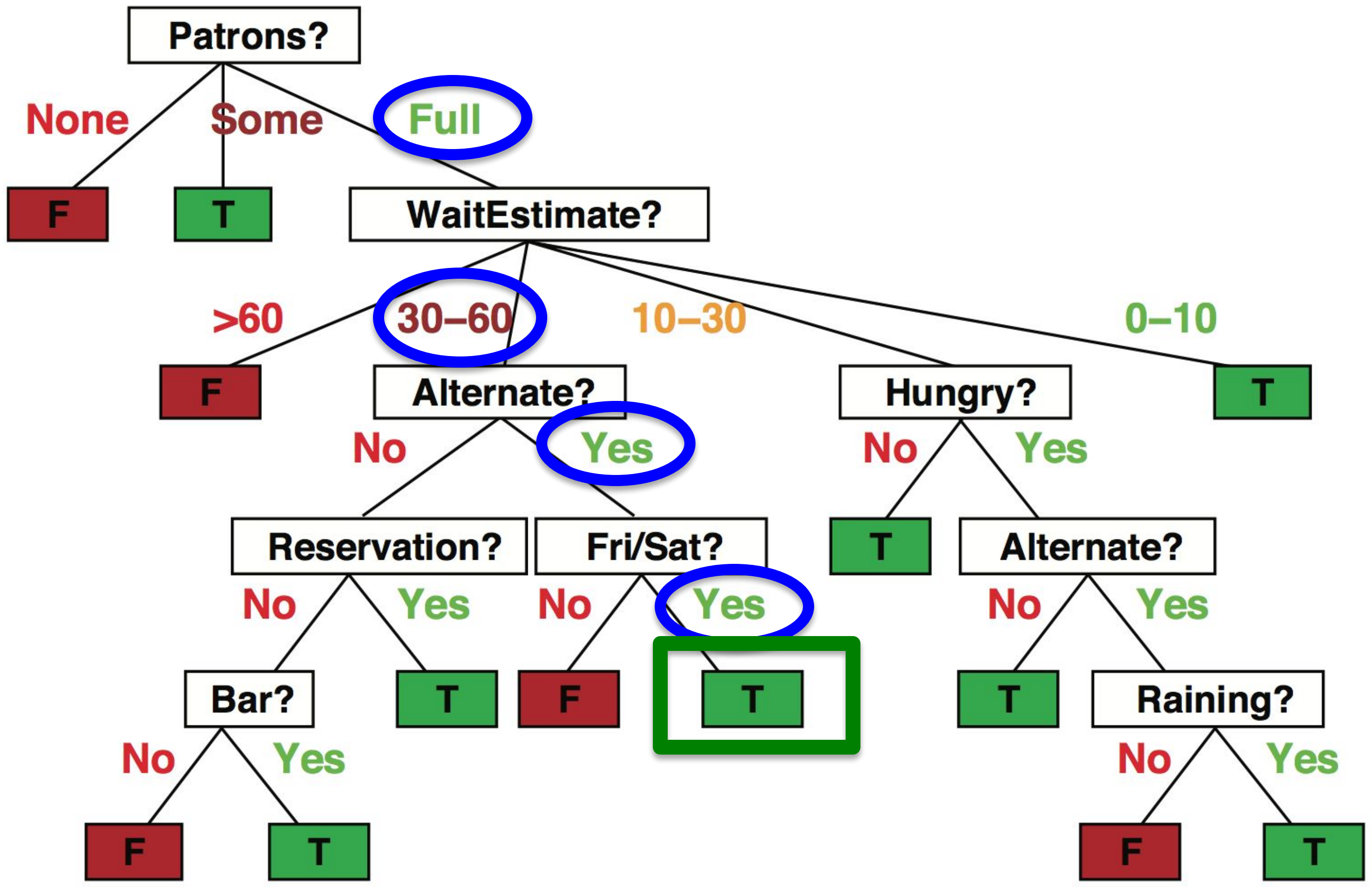
- Popular representation for classifiers
  - Even among humans!
- I've just arrived at a restaurant: should I stay (and wait for a table) or go elsewhere?



Decision trees

- It's Friday night and you're hungry
- You arrive at your favorite cheap but really cool happening burger place
- It's full up and you have no reservation but there is a bar
- The host estimates a 45 minute wait
- There are alternatives nearby but it's raining outside

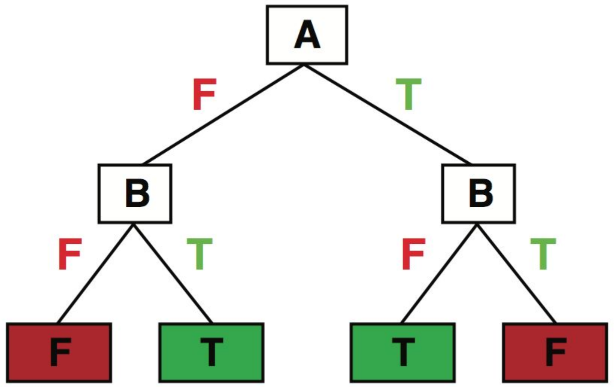
Decision tree ***partitions*** the input space, assigns a label to each partition



Expressiveness

- Discrete decision trees can express ***any function*** of the input in propositional logic
- E.g., for Boolean functions, build a path from root to leaf for each row of the truth table:

A	B	A xor B
F	F	F
F	T	T
T	F	T
T	T	F



- **True/false**: there is a consistent decision tree that fits any training set exactly
- But a tree that simply records the examples is essentially a lookup table
- To get generalization to new examples, need a compact tree

Quiz

- How many distinct functions of n Boolean attributes?
  - = number of truth tables with n inputs
  - = number of ways of filling in output column with 2^n entries
  - = 2^{2n}
  - For $n=6$ attributes, there are 18,446,744,073,709,551,616 functions
    - and many more trees than that!

Hypothesis spaces in general

- Increasing the expressiveness of the hypothesis language
  - Increases the chance that the true function can be expressed
  - Increases the number of hypotheses that are consistent with training set
    - => many consistent hypotheses have large test error
    - => may ***reduce*** prediction accuracy!
- With 2^{2^n} hypotheses, all but an exponentially small fraction will require $O(2^n)$ bits to express in ***any*** representation (even brains!)
- I.e., any given representation can represent only an exponentially small fraction of hypotheses concisely; no universal compression
- E.g., decision trees are bad at "k-out-of-n" functions