COMPUTER SCIENCE
INDIANA UNIVERSITY
School of Informatics and Computing
Bloomington

Machine Learning

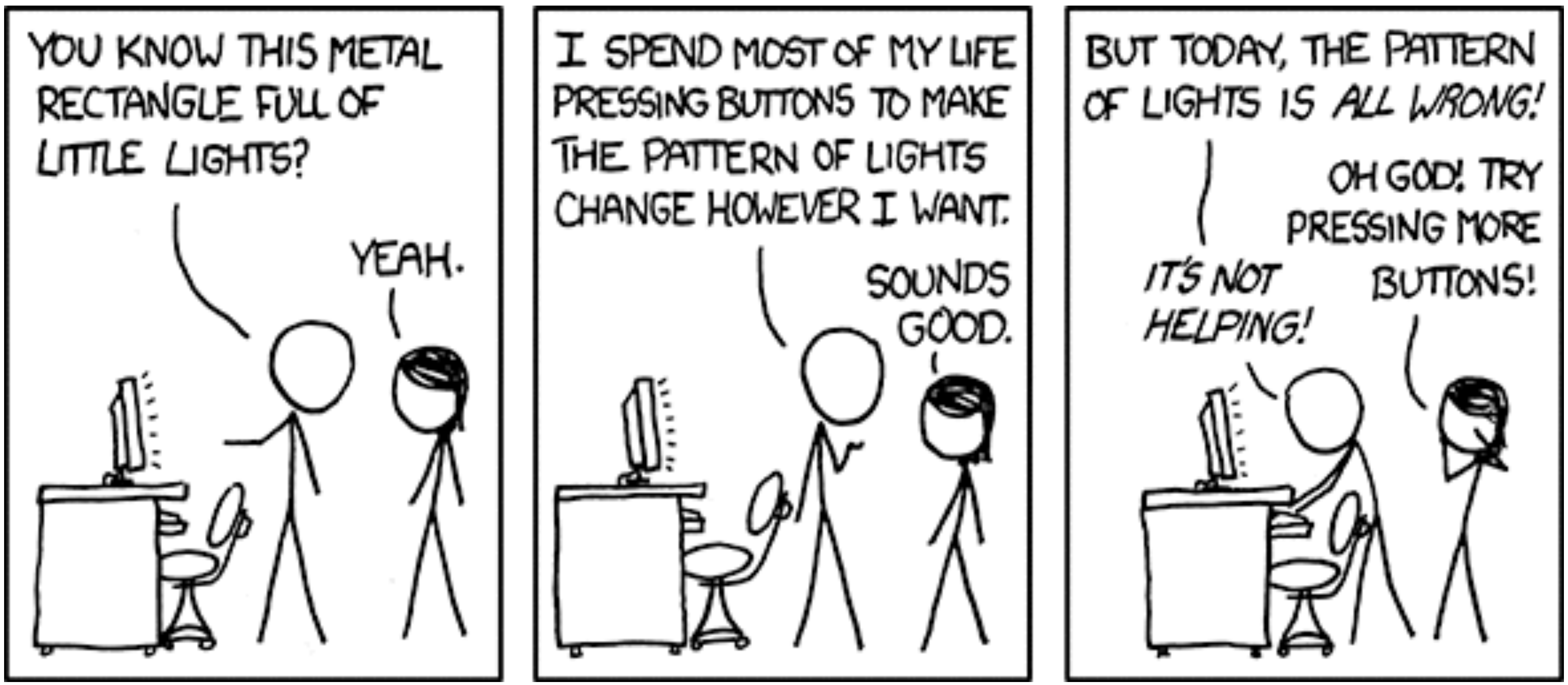
CSCI-B 555
Fall 2016
Martha White

What is this course about?

- The world is full of information and data
- Much of that data is noisy and has an element of uncertainty
  - We have incomplete knowledge of the environment
  - Actions of other actors not provided
- **Goal**: understand machine learning algorithms to analyze this data by deriving them from the beginning
  - our focus will be on prediction (on new data)

What is this course about?

- **One line goal:** demystify the seemingly huge collection of machine learning algorithms by exploring their mathematical foundation



What is this course about?

- The focus is on fundamental data analysis/statistics concepts, and not necessarily on application of these algorithms
    - though you will get to do that too

- Application of algorithms is simpler when you understand their development and underlying assumptions

- Overall goal: understand how to use (heaps of) data to make predictions about novel events, including
    - what assumptions to make and how to formalize the problem
    - how to derive algorithms for your problem
    - ascertain your confidence in the predictions (i.e., evaluate your approach)

Basic information

Class meets:
Time: MW 4:00pm – 5:15pm
Place: Jordan Hall, Room A-100

Instructor:
Martha White
Office: Lindley Hall 401E
Email: martha@indiana.edu
Web: http://homes.soic.indiana.edu/martha/

Office Hours:
Time: T 2:00pm-4:00pm (is this a good time?)
or by appointment
Place: Lindley Hall 401E

Class Web Site:
https://iu.instructure.com/courses/1560796

Associate Instructors (Teaching Assistants)

Inhak Hwang
Rakish Kumaraswamy
Andrew Patterson
Erfan Sadei Azer

Office Hours:

Times:

T 3:30pm-5:00pm (is this a good time?)

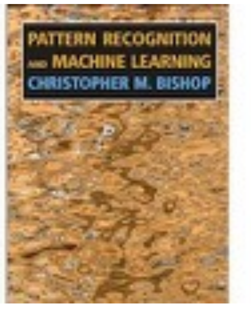
W 1:00pm-2:30pm (is this a good time?)

Place: Lindley Hall 401 open area

Textbook information

- **Main notes** provided in Canvas
  - written by Predrag Radivojac and myself (about 130 pages)

- **[Optional] reference material/recommended readings:**
  - Pattern Recognition and Machine Learning
    by C. M. Bishop, Springer 2006.
  - The Elements of Statistical Learning
    by T. Hastie, R. Tibshirani, and J. Friedman, 2009



Marks

- 40%: Assignments (4), a mixture of mathematical and programming exercises (code provided in python)

- 15%: Quizzes (3)

- 35%: Final exam, December 16, 5:00 p.m. - 7:00 p.m.
    - PhD students can choose to do a project instead. If so, you must come discuss it with me within the first 3 weeks

- 10%: Thought questions (3), show that you're reading and thinking about the material

More on grading

- Top performers will get A+
  - This course is not curved; I decide the letter thresholds (Note that I don't enjoy having to assign you a grade, but it helps you learn)
- Previous course distribution: 1 C, 3 C+, 4 B-, 6 B, 9 B+, 4 A-, 9 A, 7 A+
- Final mark includes only top 3 best assignments for each student
  - i.e., your worst assignment mark will not be included in your final mark
- No arguing for marks with AIs
  - should only ask: "Can you help me understand this question?
  - Even if you get 10% improvement on an assignment, at best this translates into 1% for the final grade. It is a waste of your (and everyone's) time.

Submission policy

- Homework assignments are due on the date specified in Canvas
  - always Wednesday, at 11:59 p.m.

- Submit assignments and thought questions in Canvas

- Late assignments will be accepted according to the following rules

– **points**	**(on time)**	**recommended!**
– points x 0.9	(1 day late)	not recommended!
– points x 0.7	(2 days late)	
– points x 0.5	(3 days late)	
– points x 0.3	(4 days late)	
– points x 0.1	(5 days late)	
– 0	(after 5 days)	

- There can be legitimate issues; if something is wrong, talk to me early

More about assignments

- **All assignments are individual:** no discussing questions
  - The assignments are feasible on your own; you will learn much more and get more out of the course when you accomplish this
- Acknowledge sources (websites, books) in your documents
  - can be typed up or legibly handwritten assignments are in LaTex; I have given you latex files on Canvas, for each assignment

Academic honesty

- I have to report every cheating incident to the university
  - If you're feeling so overwhelmed that you need to cheat, then come talk to me and be honest and we can find a solution
- Code of student rights, responsibilities and conduct: http://www.indiana.edu/~code/
  - e.g. Students are responsible to "facilitate the learning environment and the process of learning, including attending class regularly, completing class assignments, and coming to class prepared".
- You are in graduate school, so I hope one of your priorities is to learn and become a more independent thinker

My expectations

- Basic mathematical skills
  - calculus
  - probabilities
  - linear algebra
- You are hardworking and motivated to learn (machine learning)
- You are motivated to succeed in class
- You are motivated to think beyond the material and ask open-ended questions (one of the goals for graduate school)

What to expect from this course

- I know you have expectation of me too
  - I will try to be transparent in marking and course choices
  - I am here to help you learn; I will treat you with respect and listen thoughtfully to your questions (I love to answer questions and give advice!)
  - Feel free to give me feedback (e.g., Miss Martha, you are talking too fast)
  - I will provide an mechanism for anonymous feedback
- This course will be quite mathematical, with derivations of details
  - this is absolutely necessary, and will make you much more skilled in ML
- By the end of the course, you should have a good grasp of fundamental concepts in ML and algorithm derivation for ML

Thought questions

- Graduate school is about thinking and asking questions
  - maybe surprisingly, research is actually about asking good questions
  - a question does not have to have an answer, but it can be thought-provoking or make you think about a topic differently
- "Thought questions" correspond to (short) readings in the notes, and should demonstrate you've read and thought about the topics
- There are no stupid questions (so ask any questions in class/office hours/email), but "thought questions" are to demonstrate insight and so I have some requirements

General format for thought questions

- (1) First show/explain how you understand a concept
- (2) Given this context, propose a follow-up question
- (3) Propose an answer to the question, or how you might find it

Examples of “good” thought questions

- After reading about independence, I wonder how one could check in practice if two variables are independent, given a database of samples? Is this even possible? One possible strategy could be to approximate their conditional distributions, and examine the effects of changing a variable. But it seems like there could be other more direct or efficient strategies.

- PCA encodes a simple linear relationship between the data and underlying subspace. Why is PCA so widely used? It seems simple and I would not expect it to be able to encode complex properties. Potentially its simplicity is an answer.

Examples of “bad” thought questions

- I don’t understand linear regression. Could you explain it again? (i.e. a request for me to explain something, without any insight)

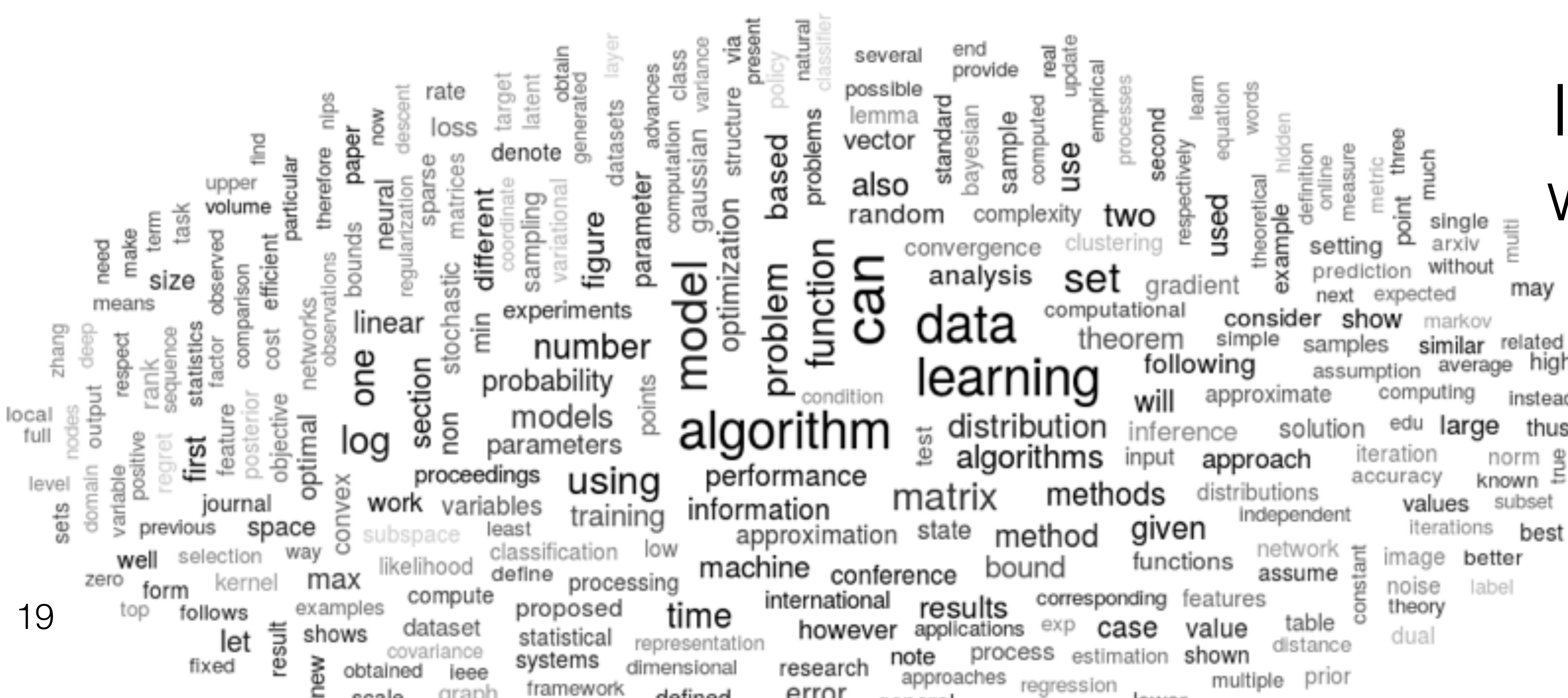
- Derive the maximum likelihood approach for a Gaussian. (i.e., an exercise question from a textbook)

- What is the difference between a probability mass function and a probability density function? (i.e., a question that could easily be answered from reading the definitions in the notes)

- How are Boltzmann machines and feedforward neural networks different? (i.e., again a definition)
    - But the following modification would be good: “I can see that Boltzmann machines and feedforward neural networks are different, in that the first is undirected and the second directed. How does this difference impact modeling properties and accuracy of estimation in practice?”

What is machine learning?

- We could label it in many different ways including
  - Data-driven approach to artificial intelligence
  - Applied statistics
- Some keywords associated with machine learning
  - Prediction, probability, samples, data, function, optimization, stochastic, …



ICML 2015
word cloud

Where did machine learning come from?

- Let’s first step back to the goals of artificial intelligence
- Many original AI approaches were expert-based
  - logic approaches for theorem proving
  - expert systems
- Machine learning arose as a data-driven approach to solve artificial intelligence problems
  - why the shift? increased computation, availability of data and efficacy of data driven approaches (largely driven by availability of data)

What are the ultimate goals?

- The focus for many ML researchers has shifted from AI towards generally solving important (practical) problems
  - computer vision, speech recognition, clustering, modeling temporal data, …
- This includes a focus on understanding intrinsic properties of a learning problem
  - is it difficult to learn (e.g., NP-hard)?
  - how can it be formulated in a precise way? (e.g., explicit probabilistic assumptions, preference for “simpler” hypotheses)
  - how many samples are needed to learn the model (epsilon) accurately?
  - how well does the learned model generalize to new samples?

How do learning problems differ?

- They can be categorized across several dimensions
  - **Control versus prediction:** though a control algorithm will likely use predictions to improve decision-making (e.g., reinforcement learning)
  - **Supervised and unsupervised:** supervised learning is for prediction, unsupervised learning is usually for visualization or representation learning; some algorithms combine these two important components
  - **How they are used:** empower decision-making of end user OR autonomously control system
  - … there are more, but these are some main ones

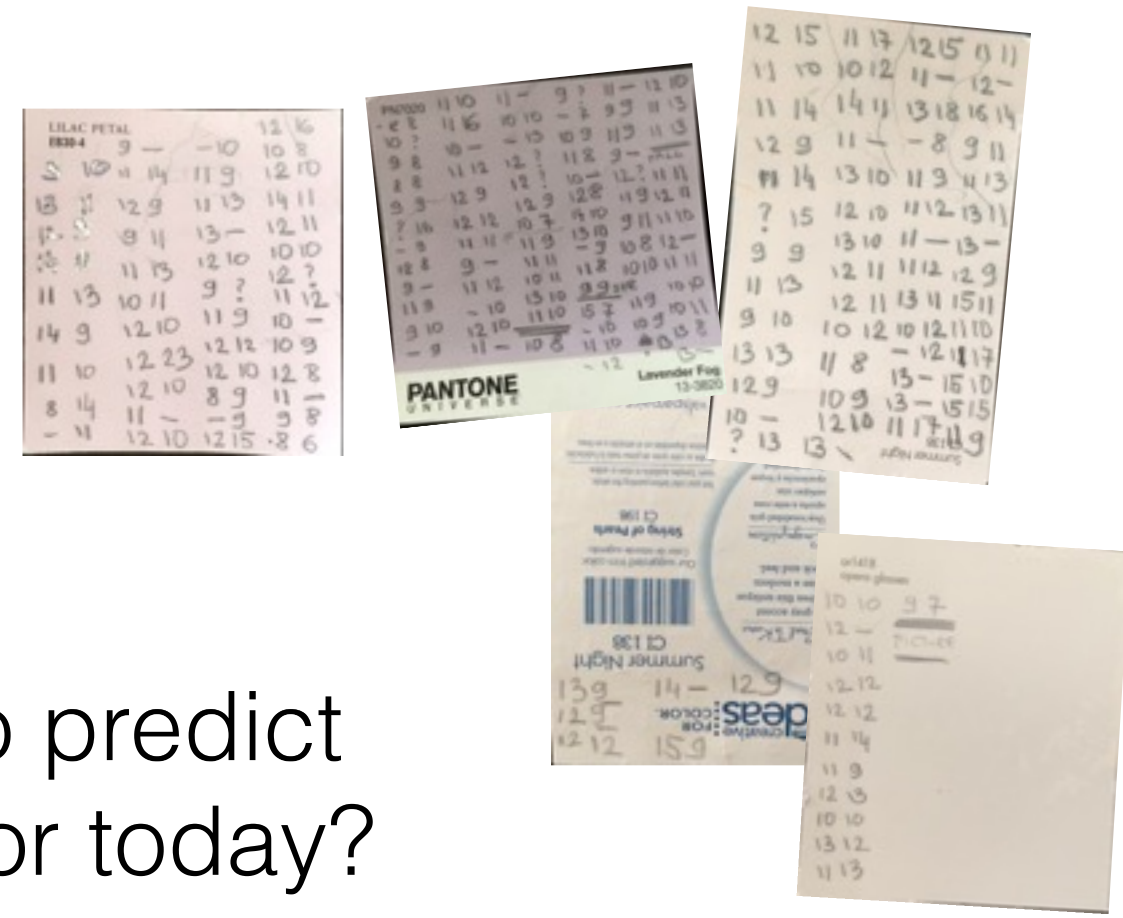
How do the algorithms differ?

- **Algorithms** also differ in many ways, even for similar problems
    - Process data incrementally (as stream) or in batch
    - Low computation (or memory) versus heavy computation (or memory)
    - Data efficient (needs only a few samples to learn a good model)
    - Consistency: with more samples, model approaches “true” model
    - … other common algorithmic distinctions, such as approximate vs. exact, or randomized vs deterministic

Let’s look at some fun examples!

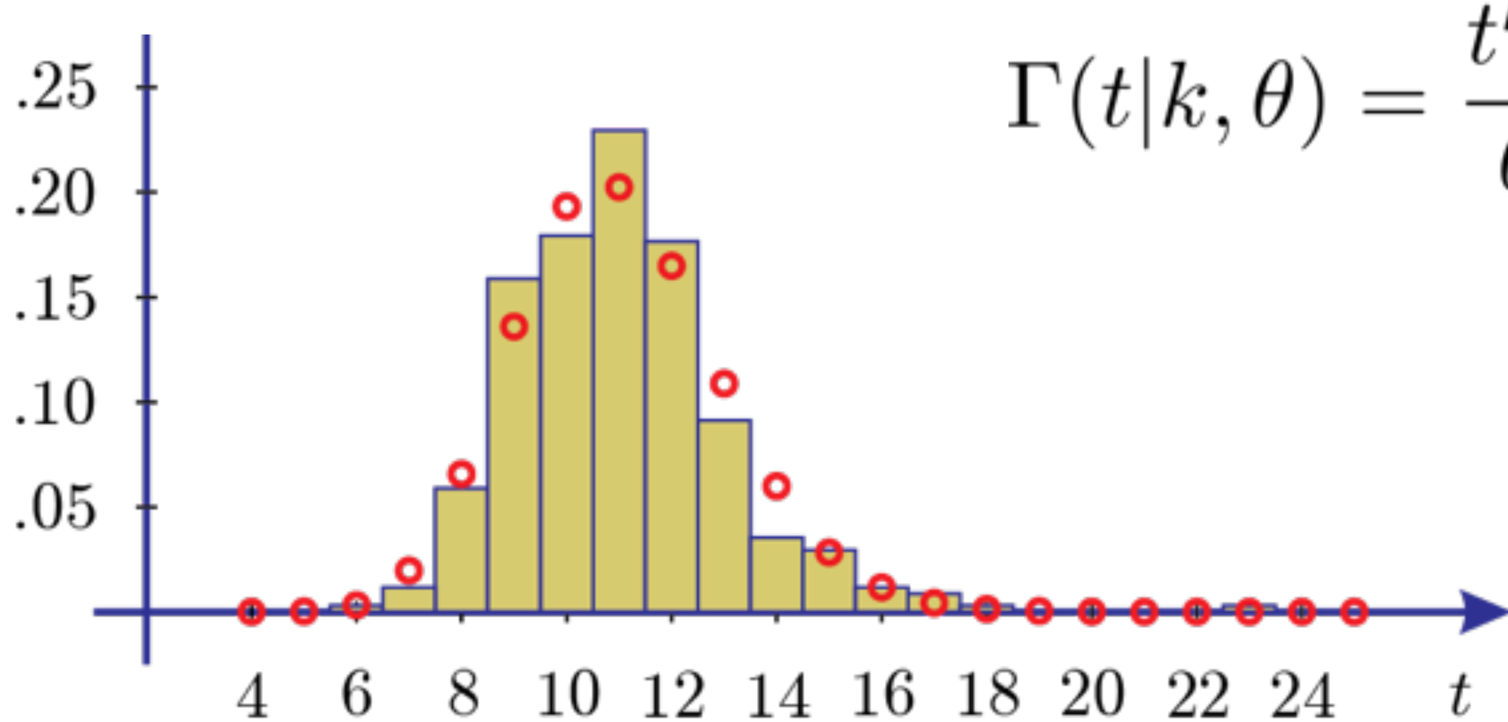
- Commute times (independent identically distributed (iid) data)
- Weather prediction (temporally connected data)
    - machine learning is often used for time series, but in the specific case of weather, mostly expert models appear to be used (for now…)
- Octopus arm simulator (machine learning for control)
    - we will not look at control algorithms; however, their development uses the fundamental concepts in this course

Commute times

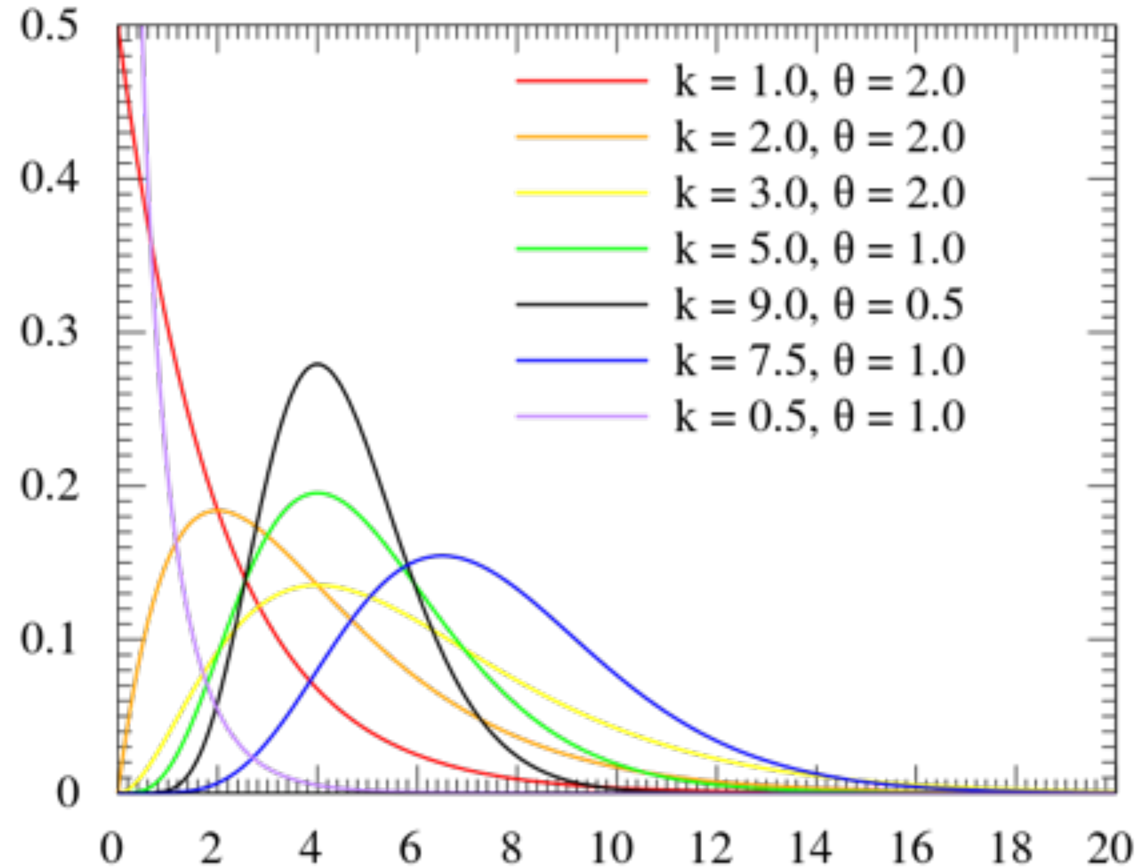


How might you try to predict your commute time for today?

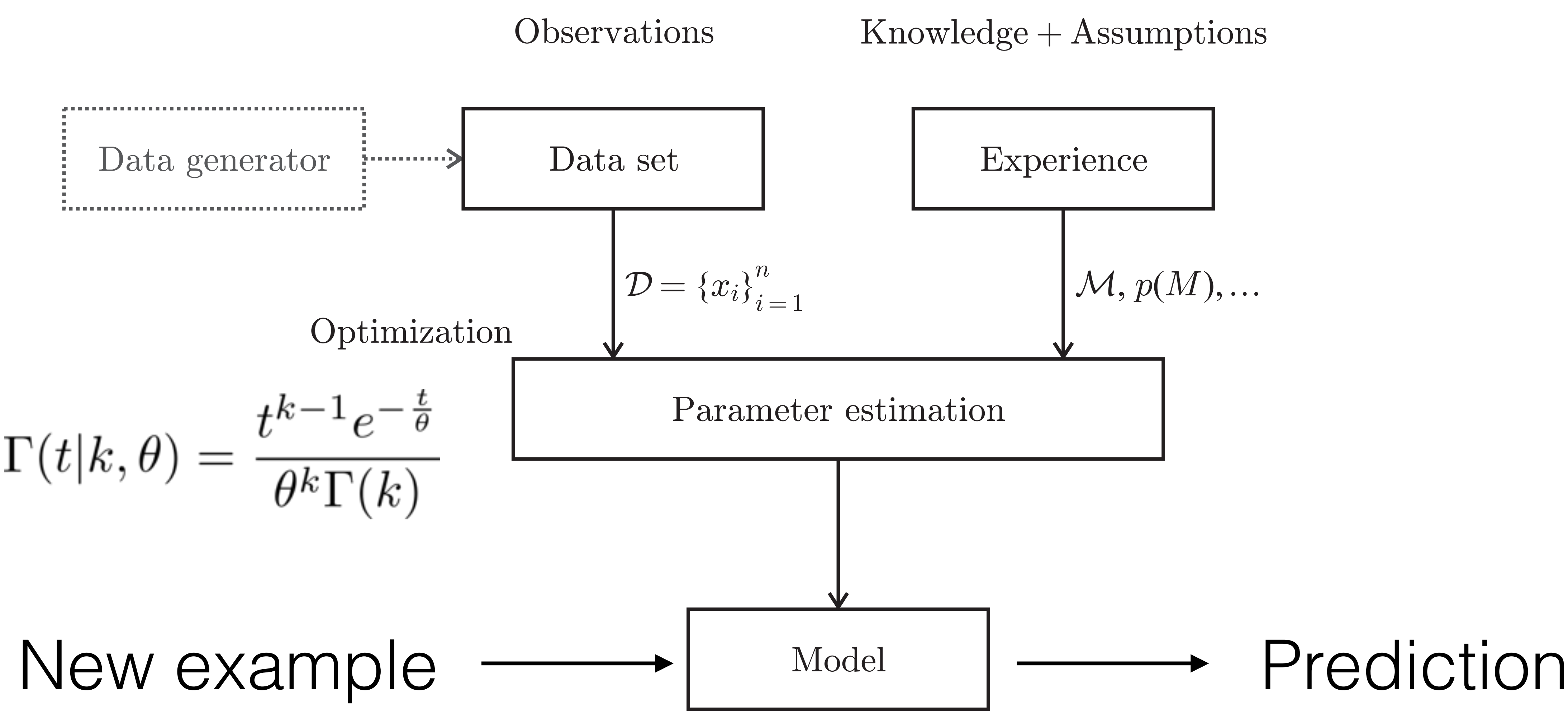
Commute times (2)



$$\Gamma(t|k,\theta) = \frac{t^{k-1}e^{-\frac{t}{\theta}}}{\theta^k\Gamma(k)}$$

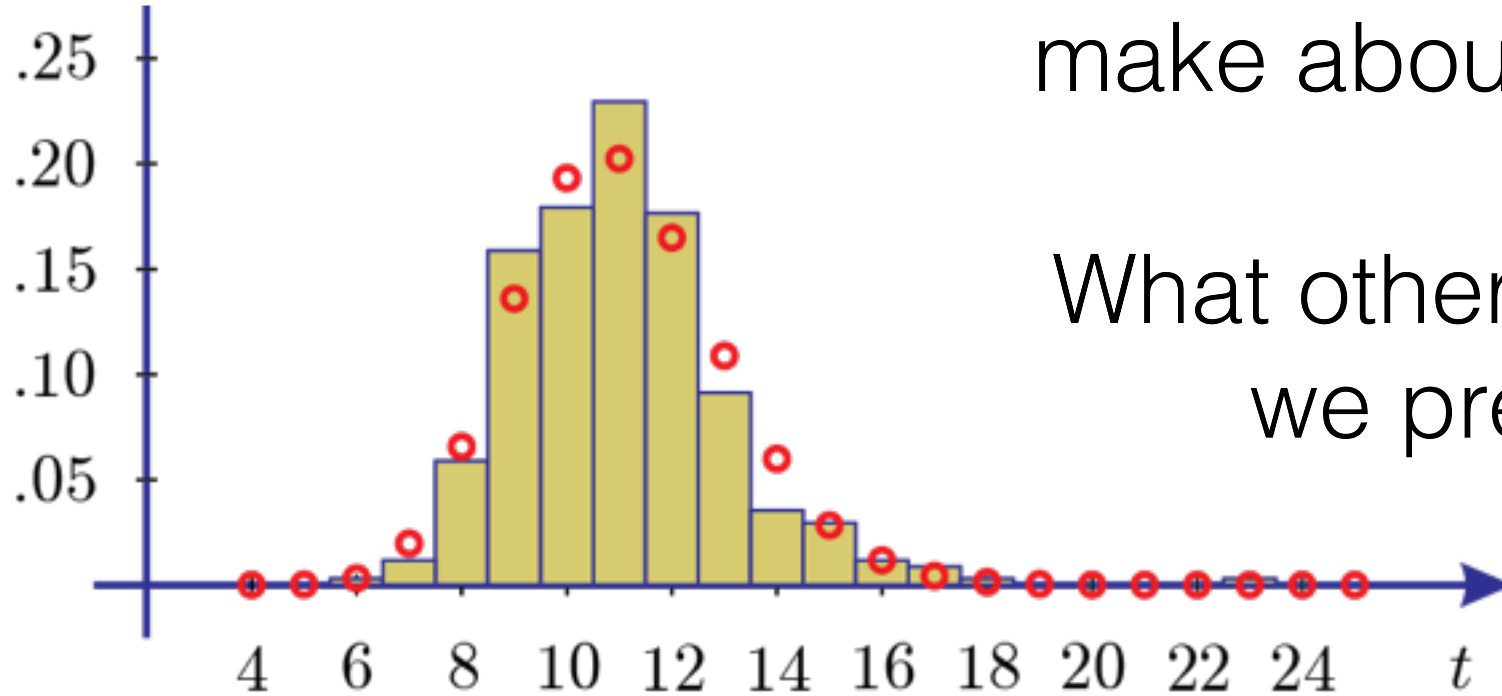


Summarized flow



$$\Gamma(t|k,\theta) = \frac{t^{k-1} e^{-\frac{t}{\theta}}}{\theta^k \Gamma(k)}$$

Commute times (3)

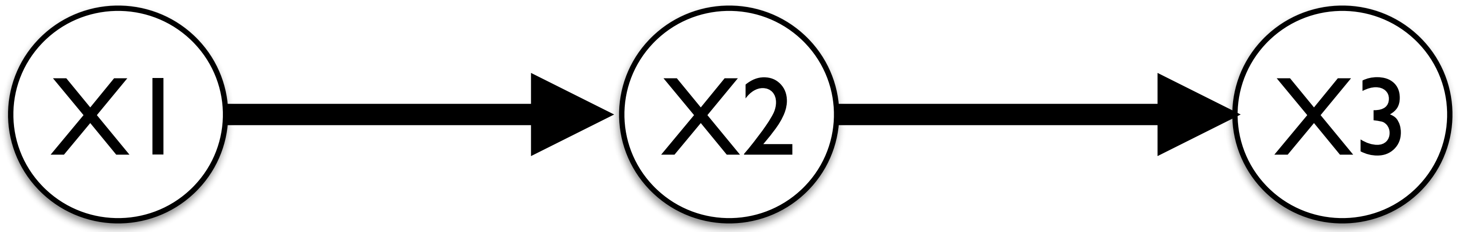


$$\Gamma(t|k,\theta) = \frac{t^{k-1}e^{-\frac{t}{\theta}}}{\theta^k\Gamma(k)}$$

What assumptions can we make about the data?

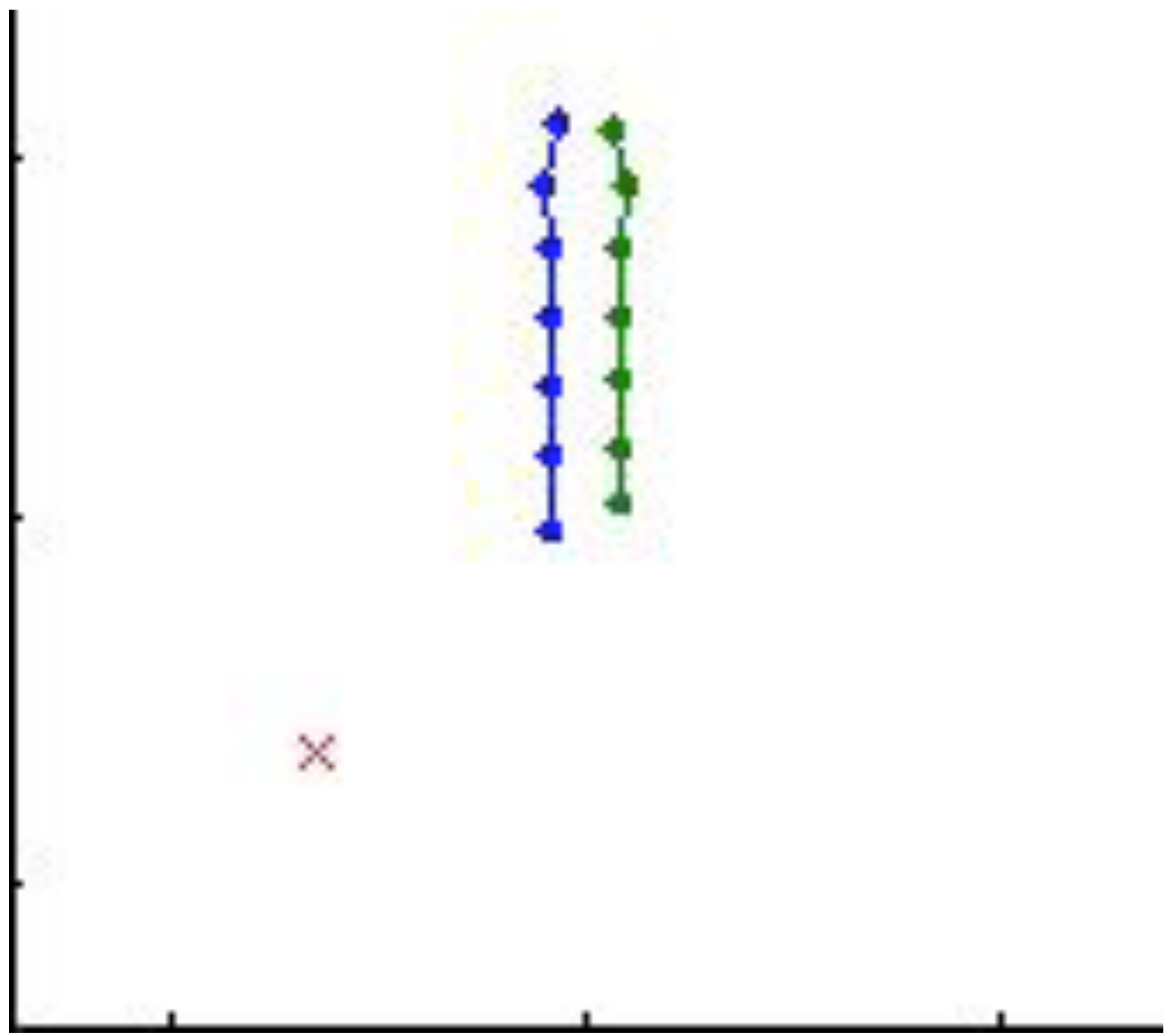
What other ways can we predict?

Weather prediction (time series)



- Imagine we want to predict the probability of rain tomorrow, 2 days from now, 3 days, …
- One common strategy for time series is to use the last p points as features to predict the next point, 2 points into future, etc.
- What other strategies can you imagine?
- How do you predict a probability value, rather than say a binary value (0 or 1) or a real value?
    - Hint: these are things we will learn

Octopus arm (control)



Uses of machine learning in the real world

- Character recognition for mail addresses (as early as mid 90s)
  - 30% accuracy in 1997 to 88% accuracy in 2004
- Spam filtering p(email = spam | information about email)
- Netflix challenge (matrix completion)
- Speech recognition (deep learning)

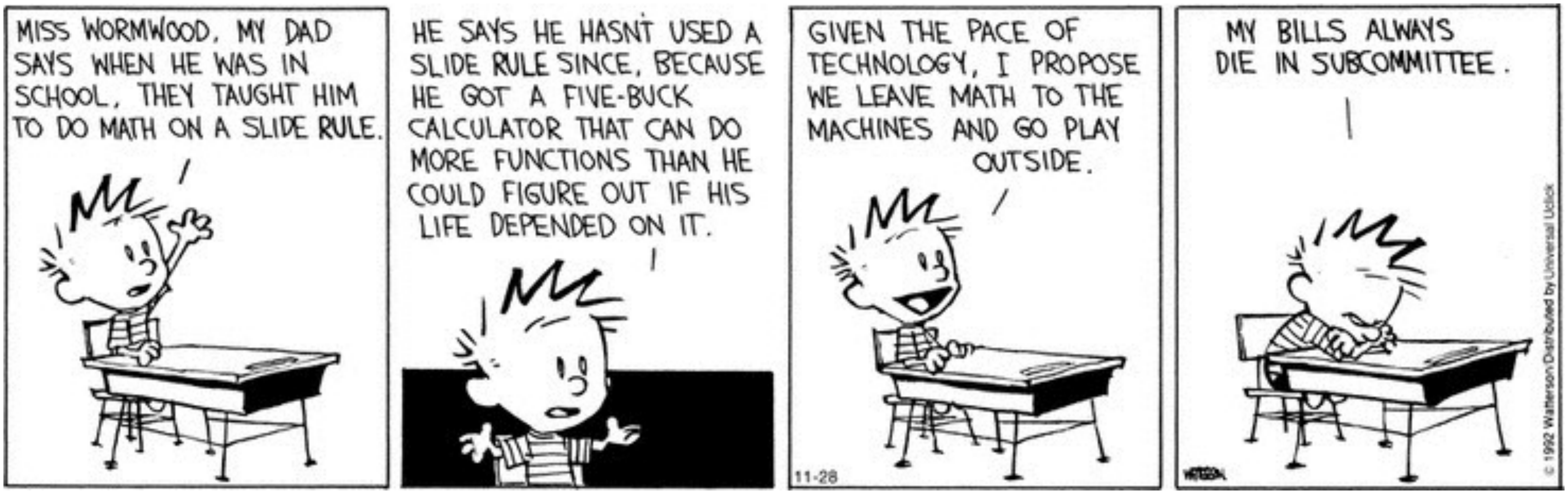
When is it appropriate to use ML? What is not using ML?

- Many real-world systems are starting to have some pieces specified using ML, and others solved using other approaches
  - often other components are expert-based

- Many systems use data processing and statistics. Is this ML? Is it learning? Can we use ML everywhere now?

- The best distinction might be your goals, some of which might be
  - use probabilistic modeling to solve a specific problem
  - design an agent that can learn continuously from a stream of data
  - build a tool that predicts stock market prices
  - gather statistics on a population of individuals, obtain regression models for understanding that data

Next: crash course in probability



- Probabilities underly much of machine learning
  - enable precise modeling of uncertainty
- Understanding assumptions and how to build extensions is key for effectively using machine learning algorithms

Topic overview

- Parameter estimation and prediction problems (Chapters 2 and 3)
    - the core background for modeling in machine learning
- Linear regression (Chapter 4)
- Generalized linear models (Chapter 5)
- Linear classifiers (Chapter 6)
- Representations for ML (Chapter 7)
    - help make linear predictors more powerful (e.g., neural networks)
- Statistical learning theory and empirical evaluation (Chapter 8)
- … and any other topics listed on the syllabus if we have time (e.g., boosting)

Anticipated questions

- Where can I find background exercises/questions?
  - Try machine learning books (e.g., Barber's book)
  - Try applied statistics textbooks (e.g., All of Statistics)
- The course is too slow / too fast
  - If too slow, come talk to me and I'll show you how to get more from it
  - If too fast, don't be too frustrated; slowly the information and way of thinking start to make sense.

Anticipated questions

- Why are we learning such simple models? Isn't the modern approach neural nets and we should start there?
  - The foundational material is key for properly understanding more complex models. As you will see, building up from the foundation is critical to understanding neural nets. Many people do not understand that backpropagation is gradient descent, the choice of activation functions, etc.
  - Without an in-depth understanding, you will not be able to use machine learning effectively for your novel setting, and may even make terrible modeling decisions

Anticipated questions

- Why aren’t we programming more? I don’t need to learn about linear regression, I can just use packages.

- Quote from previous student: “One thing I would like to strengthen is the importance of knowing the mathematical details of how to deduct each model. Some people might think it is too tedious to know, but I have done projects on modifying training algorithms, like adding priors or changing the lose functions, and it requires me to know those details so that I know what I am supposed to do. Projects related to basic models, like linear regression, logistic regression, naive bayes, decision tree and support vector machine, require me to truly understand the models. Although there are various packages to choose and one don't need to implement a model from scratch, knowing the details helps in parameter tuning and feature engineering and it also makes me creative in finding new ideas.”

Anticipated questions

- My math skills are poor. What should I do?
    - Math is just a tool/language. Practice and become more comfortable with this language. A common pitfall is to try to intuit all the math; I recommend against this. For example, try to learn the notation behind probability first, before getting a strong intuitive grasp, and once you are more comfortable with the notation, then start searching for intuition

- I’m rusty at programming. Am I going to fail?
    - The amount you program is limited. I provide python code to read in data and do basic learning on that data. You will simply have to modify this code, likely amounting to at most 500 lines of code.

Anticipated questions

- I was hoping we would learn about topic x, but it looks like it is not listed. Can we learn about topic x?
  - With the foundations from this course, you will much more easily be able to go learn about more advanced topic x
  - Otherwise, particularly later in the course, come chat with me and we can discuss topic x