LECTURE #1

LECTURER: TOM GRIFFITHS (PSYCHOLOGY DEPT., U.C. BERKELEY) SCRIBE: KIRAN VODRAHALLI

Contents

1 Introduction

What I do in my research is I try to understand how people do inductive leaps: figure out causal relationships, interpret words and sentences, discover meaningful features of objects, learn functions, languages, and concepts from limited data. These are examples of things people tend to do. These are problems for which human cognition still sets the standard. If we can figure out how people do these things, we can improve machine learning methods.

The general approach I take in my research is I try to build those links. What is the computational problem involved in some aspect of computation, then using math to finding the optimal solution, and then comapring human cognition to the solution; then you can find if it is a good model or not. We directly relate human and machine learning.

What makes people good at solving these problems like learning new words is having biases that guide us towards the right solutions. We can measure those biases and implement them in machine learning systems.

Human learning also gives us problems that we would like machines to be able to solve as well. This research program addresses levels of analysis: Different ways we can ask questions about human computation.

1. What is the goal of the computation, why is it appropriate, what is the logic of the strategy by which it can be carried out?

2. What is the representation for the input and output, and alg for transformation.

3. How is the representation and alg realized physically?

We will take the abstract links between human and machine learnings and ask questions about cognitive processes that humans use to solve problems. There are two views of human mind: 1) humans are good at

solving these problems - we can solve problems in world around us; 2) humans are bad at solving problems, we have systematic bias, there are lots of errors we make, so maybe we are bad models upon which to base machine learning systems.

2 Rationality

2.1 Explaning inductive leaps

What is the structure of the computational problem being solved? What is the right mathematics? A lot of work in this domain is based on the premise that there is not new mathematics we need, but rather that we can just use Bayesian inference. Bayesian statistics provide a formalism we can use to suggest what leaps of human beings might like (Recall Bayes rule: $\mathbf{P}\{h|d\} \sim \mathbf{P}\{d|h\}\mathbf{P}\{h\}$). Really, this is an inductive rule. The key assumption this makes is that we encode our beliefs as probabilities and updating them based on data. Here are some examples:

1. Causal learning (human learning) - Graphical models (machine learning). See Griffiths and Tenenbaum, 2005. We can describe causal learning by estimating parameters for a simple graphical model. All of the previously proposed models can be represented by this graphical model (Background and Cause with weights w_0, w_1 feed into the event E). So how can we formalize the notion of causal learning beyond these very simple cases. We can also ask more sophisticated questions about causal learning: What are the expected values of the parameters, and what the prior distributions look like (estimated empirically with MCMC). People have an expectation that causal relationships should be strong. We can therefore get clear pictures of what people's assumptions about causality are like. The groundtruth for discovering causal relationships is in fact the human suggestion. So you can use these priors in order to do a better job of identifying causal relationships.

2. Iterated learning. Implement a Gibbs sampler with $p(d, h)$ as its stationary distribution and hence converges to the prior. See Griffiths and Kalish $2005, 2007$. We can apply to gene expression as well. We estimate parameters and use them to simulate prior distributions.

3. We can use a similar kind of approach to Memory (human learning) and Information retrieval (machine learning). For instance, you can use topic models to express document similarity (probability distributions over topics). We made models of human semantic memory as a representation of semantic content. These might work better than vector space approaches etc, and result in better methods for estimating topic models. Something else we might consider is web search (identify a web page based on a search term). Griffiths and Tenenbaum 2007.

4. Learning features (human learning) - Dimensionality reduction (machine learning). For instance, I can show you a picture, and the things you identify as the meaningful parts of that picture might be different from if I showed you exactly the same picture in a different set of objects. We would like to be able to explain how we form representations of meaningful features of things in the world. We want to balance structure and flexibility, but we do not want to constrain in terms of number of features. We can use nonparametric Bayesian statistics (Indian buffet process allows us to capture fact that features may vary - Griffiths and Ghahramani, 2005). This kind of model does a pretty good job of capturing different kinds of features. Subsequent work has been done in computer vision and machine learning. Indian buffet process is a prior people can use: It is a good prior to use in data without making too many assumptions. The key point is that we end up with a list of things people do and a list of methods from machine learning and statistics.

5. Cateogrization - Density estimation

6. Language learning - Probabilistic grammars

7. Experiment design - Inference algorithms

One side is the NIPS side, one side is the other side (go to both conferences to get solutions from one to the other). But all this depends on the notion of rationality.

3 Heuristics

The Great Rationality Debate: Is it reasonable to characterize human cognition as fundamentally rational? Unbounded computation being applied to solving problems in exactly the right way. Psychology does not believe that rationality is the way humans are. Here is the classic book: Judgement under uncertainty: Heuristics and biases. In this research program, solving problems in probabilistic inference is hard. People aren't doing that, so let's try to identify heuristics people are using to solve problems. The message taken away from that research is that because of those biases, people are irrational. Thus we deviate away from looking at how people solve those problems. We are getting more and more depressed about human cognition. There are two classic heuristics we will talk about.

3.1 Anchoring and adjustment

The idea is that people start with a more familiar example (an anchor) and adjust away from it (leading to bias). Start out with what Earth's orbit is, then use that to estimate Mars (adjust away from Earth).

So if you give something to anchor on, people will take that.

3.2 Availability of extreme events

People over estimate probability of events that come to mind easily. If something is easy to generate a model of, you have too high an estimate of the probability, particularly for extreme events. Are you going to go scuba-diving: You probably won't think a shark will happen. People overestimate terrorist attacks, etc.

3.3 What makes a good heuristic?

Can heuristics be good? Second, are all heuristics 'kludges'? Or are they things w.r.t. criterion of being a good heuristic actually give us a good match to the problems we are trying to solve?

Can we define a formal framework for exploring heuristics, as for rational models? Optimization is inherent in the rational models, whereas exploring heuristics seems much less specified. We end up with a grab-bag of heuristics, we would like to systematize and know when certain heuristics are good. That is what we will do in the rest of the talk

4 Cost of Computation

We will reconcile the views of rationality and heuristics in this section.

When we classically talk about rationality, we don't typically talk about computational resources available for agents. However, we, just like the machines, are computational devices, and we have limited capacity and time. Thus there is a cost associated with thinking. We want to tradeoff that cost with the goodness of our answers. We have joint work with Lieder, Goodman 2015. Starting at the computational level, where the problems are solved, down to the algorithmic level. Let us characterize what constitute optimal algorithms with respect to computational cost of algorithms.

We get some information from the world, and we can also get information by running a simulation. This might give you further information you can use to guide your actions. We can apply same rational actor model such that actions are not just actions on the world, but actions on the computer. We can gain further information from computations, which can inform our decisions. No longer an outcome focused view of rationality, to a more process-oriented view of rationality: Executing the appropriate computations in order to trade off error from world with the cost of computation (runnign computer inside your head - rationality). "Bounded optimality" gives a characterization of how computational agents should act rationally, making use of computational resources available to them - Stuart 2015. We can simply use TIME - more time thinking, less time acting. With that simple assumption, we make a lot of progress. We can then characterize heuristics, which hits that tradeoff. Classic heuristics are in fact things we can identify as being resource rational (from our two examples before).

We can think about both classes of heuristics as a computational architecture being used to solve that problem. The anchoring and adjustment: Task: Estimate a quantity based on memory and other cues (probability distribution over values). Some kind of iterative simulation architecture (Metropolis-Hastings, MCMC) - a way to sequentially estimate and update. The cost increases linearly with number of samples (as with opportunity cost).

The Metropolis-Hastings algorithm is MCMC: Define Markov chain with stationary distribution equalling the distribution you want to sample. Thus if you run it long enough it will converge. You iteratively adjust estimates until at some point, the markov chains converge together regardless of where they started. Initial point in MCMC is 'anchor' and after simulations, we get distribution over responses. So how long should you run your MCMC for. This is not just for human beings trying to make an estimate of Mars. This is problem for humans trying to build efficient machine learning algorithms.

Machine learning researchers for statistics people: ML: Run algorithm for as long as you have to get results, statistics people run for as long as possible to find truth. We are trying to answer how long it is you should be running your algorithm if you assume there is cost associated with running it longer. The key property of this algorithm is that we require the bias of the algorithm to decay exponentially. The different between true value and algorithm value is decreasing as a function of the number of iterations you run (Lieder, Griffiths and Goodman, 2012). If there is a cost for each iteration, the optimal thing to do is to stop before your bias has decayed to some minimal level. The rational amount of bias you should accept is a function of ratio of iteration cost to error cost. The amount of bias you should take is some non-zero quantity for any setting. If you have limited time, you might want to take the hit in terms of bias. We can use this model to give us predictions about what people might do in these sorts of tasks.

Using sampled data distributions of answers to bias questions, we estimate the number of steps and relative step size across things to find out what the rational point is you should stop your running of MCMC. And this led to good prediction of when people actually stopped.

Anchoring and adjustment:

1. Cognitive load, time pressure, alcohol reduces the amount of adjustment they do: increases computational cost.

2. Bias increases with anchor and extremity.

3. Amount of uncertainty increases anchoring (more uncertainty of algorithm, tightens up distribution (stay in same place, increases convergence rate as well)).

4. Knowledge can abolish bias.

As for availability of extreme events, the task is to estimate expected utility of an action. As for architecture: generate weights samples of possible outcomes. Cost increases linearly with sample size.

For Russian Roulette, based on our original model, you must simulate Russian Roulette around 51 times to decide to not play russian roulette. You can use something called importance sampling to correct for this

excessively large simulation cost. You might have some nasty distribution you want to approximate. Instead reweight samples from a nice distribution. Assign weights proportional to p/q, then normalize weights. Then we have an approximation to original distribution. We want to minimize variance, which is analytically solveable. But this result is biased: with small samples, we will over-represent extreme events.

The optimal estimator is an estimator that over-estimates extreme events. We asked 100 people to consider 37 life events, give an estimate of frequency of events and how bad they are. Lethal events are extreme, mundane events are not. Mundane events are estimated accurately, where stressful events are overestimated, and causes of death are drastically over-estimated. This model also reproduces:

1. Fourfold pattern of risk preferences (Tversky and Kahneman 1992)
2. etc. (others)

Another problem comes out of this framework. So what are the scientists doing to produce science? The question is how are we able to make rational choices about what computations to perform without producing additional computations? There is a solution called rational metareasoning. We assume the mechanism for choosing algorithms is 'relatively done'. For each strategy we might follow, what I do is estimate how long it will take me, and how long that heuristic will be. You can build a regression model with some features of the problem to estimate how long it will take me to solve a particular problem. This is pretty good at choosing adaptive algorithms. Very different from previous work in psychology on strategy-selection (model-free reinforcement learning). Here, you are basically building a model for how long it will take to execute a computation, and then using that to choose which computation to execute.

4.1 Choosing sorting algorithms

You get people to gain experience about using sorting algorithms individually first, and then give them the option to use whichever sorting algorithm they want. To analyze, you examine some features: pre-sortedness and length. It turns out that humans are really good at learning when to use a specific algorithm, even better than computers. So modeling that process could develop better approaches for adaptive algorithms.

5 Conclusion

Resource rationality provides a way of characterizing what makes a good heuristic. Bias is an essential part of efficiently and effectively solving challenging problems. Some classic heuristics are not kludges but rather resource-rational for some architecture.

Bias is not always bad: it is just a price you pay for having limited rational resources.

So how does parallelism fit into that framework? We have been focused on serial tasks. It would be interesting to apply this framework to those kinds of questions. How do you choose which systems to use?

Some people to look at: Falk Lieder, Noah Goodman, Joe Austerweil, Jess Hamrick, Nick Hay, Ming Hsu, Dillon Plunkett, Stuart Russell, Saiwing Yeung. Check out Computational Cognitive Science Lab.