Contents

1 Introduction (Sanjoy Dasgupta)

I'll talk about 5 areas which need foundational work.

1. Teaching
2. Explanations and interpretations
3. Unsupervised++
4. Imitation
5. Semantic communication

These areas all have one feature in common: Cooperation between agents of different types, that don't know each other's insides (ex. machine and human).

1.1 Teaching

There is a spectrum of types of examples: adversarial (in online learning), random (in statistical learning), benign (in teaching). Does sample complexity dramatically improve? People have converged upon a particular model that is influential but also broken.

What is the minimum set of labeled examples needed to uniquely identify the target concept? It's a kind of description dimension. This is in relation to a specific concept class C.

Definition 1.1: Let C be a concept class and $h \in C$ be a target. $TD(h, C)$ is the smallest set of labeled instances for which h is the only consistent concept in C.

We can define $TD(C) = \max_h TD(h, C)$, or $\mathbb{E}_h TD(h, C)$.
This is geared towards finite concept classes.
This is broken because of the following problems.

1. It assumes the teacher knows the representation of the learner and the learner's concept class. Examples are tuned to the concept class.

2. The problem of selecting the teaching set can be NP-hard.

3. The predictions it gives for ideal teaching sets are ridiculous, ex. cat that looks like dog and dog that looks like cat. In practice people select examples that are far apart (the canonical cat/dog).

4. This only works for the realizable case.

What to do? What are better teaching models?

1. Who is teaching who? Human/machine teaching human/machine?

   (a) Human-human: education/cognitive science/childhood development

   (b) Human-machine: machine learning

   (c) Machine-human: intelligent tutoring

   (d) Machine-machine:
   cf. cotraining, two machines bootstrapping each other.
   GAN's teach each other to generate/discriminate (with opposite goals).

2. Avoid assuming the teacher knows the learner's represntation and concept class.

3. Interactivity: The machine could ask questions.
   Any semi-realistic model would be interactive.

4. Come up with models that gives far apart examples (Jerry Zhu).
   Ex. Let's say the learner does nearest neighbor. Suppose it is noisy nearest-neighbor. That pulls you apart.

5. Curriculum learning and self-paced learning strategies. Hierarchical learning. Simple things are learned first; then you add things.

6. Other kinds of tasks besides classification, ex. generative.

7. Restrict to an interesting domain like language.

The teacher does not have unbounded computation, but knows the concept and has a storehouse of examples gleaned from experience.

1.2 Explanations and interpretations

Ex. more than just saying you like a movie, say that you like a specific actor.

Ex. in computer vision In addition to giving a label (ex. zebra, antelope), give one-word explanations (stripes, antlers). Learn classifiers for these intermediate features as well. This taps into a potentially infinite latent space.

When feature space is high dimensional, this helps.

1. Models of explanation-based learning.

   What are the benefits of explanation-based learning.

2. Interpretable classifiers (transparency in ML).

   Output a hypothesis that scientists can understand.

   Accompany predictions with explanations. "Your loan was rejected because..."

   Decision trees used to do this automatically until people realized random forests do better.

   Use explanations to generate interpretable classifiers?

   Ex. sparse classifiers: give the support of which features you used.

1.3 Unsupervised learning++

Ex. topic modeling. Some are good, some are sliced/diced in various ways, some are garbage. Running the Gibbs sampler for longer doesn't solve the problem. This is ripe for interactive feedback of some type.

Sometimes you just literally need feedback; we want to quantify how much feedback is needed.

What type of interaction is useful algorithmically and for the human?

It could be relationships between data points, constraints based on features, etc. A practitioner would choose the algorithm and the form of interaction.

(Q: how to avoid tricks such as: To transmit finite automaton, grammar, write down grammar or automaton. Make an arbitrary convention of how to translate examples into a grammar.)

1. Improving unsupervised learning with interaction

   (a) Modes of interaction

   (b) How much interaction?

   Normally use Euclidean distance. What you want to use if people's subjective similarity scores?

2. Generalization theory for unsupervised learning

"Unsupervised learning++=Supervised learning–": Unsupervised learning is talked about a lot as lossy compression. Here, you don't have exactly the label you want, but have something that's associated with what you want. This is unsupervised++ because one can imagine this built on top of unsupervised learning algorithms.

The only results I'm aware of are nonstatistical results: ex. for clustering points, query $n \ln n$ distances rather than $\binom{n}{2}$. There should not be any n here at all, just ε and the distribution.

1.4 Imitation learning (and teaching)

One or two decades we'll be telling our domestic robots "this is how we like to make our coffee."

Imitation learning seems a tractable case of reinforcement learning. Imitation implicitly assumes a sequence of actions.

It's not enough to explain why we did this; we have to explain things we don't want to do? Littman, NIPS had a formalization.

1.5 Semantic communication

1. One paper was by Juba, Sudan, Goldreich. Ex. You don't know the language. What protocol can you execute? The answer is disappointing: try everything.

2. Percy Liang: A computer is in charge of blocks. You want to move the blocks to a specific configuration by telling the computer what to do.

   Throw in 2 constraints: compositionality of language, pragmatics (different utterances probably mean different things).

   (Pragmatics is like dropout: it helps but is not crucial. There's other things going on, which NLP takes for granted but would be interesting for theorists: ex. loglinear model.)

1.6 Other topics

1. Language learning and generation

2. Crowdsourcing. Designing proper crowdsourcing experiments. How do we learn from weak teachers (Amazon Turkers) that make errors?

   See work by Nihar Shar, Kevin Jameson (Next, with Robert Nowak).

   `http://nextml.org/`