11/13/19

8805 Learning from Data: Lecture 23

Notebooks for today: (see also mini-project IIIa)

- maximum-entropy/MaxEnt.ipynb, PDFs_from_MaxEnt.ipynb, MaxEnt_Function_Reconstruction.ipynb
- bayesian-methods-and-machine-learning/Bayesian_optimization.ipynb
- ** mini-project-IIIa_bayesian_optimization.ipynb ⇒ due next Wednesday.

- We probably won't have time to do the MaxEnt_Function_Reconstruction notebook, except maybe as an overview, but maybe someone can do it as a project.
  - Quick summary:

Consider a function $f(x)$ on $x \in [0,1]$ with $f(x) \geq 0$.
Suppose we know the moments of the function:

$$\mu_j = \int_0^1 dx\, x^j f(x) \quad , \; j = 0, \ldots, N$$

- Define $S[f] = -\int_0^1 dx\, (f(x) \log f(x) - f(x))$

Maximize S subject to the constraints of the moments:

$$Q[f; \{\lambda_j\}] = S[f] + \sum_{j=0}^{N} \lambda_j \Big(\mu_j - \int_0^1 dx\, x^j f(x)\Big)$$

⇒ find $f(x)$.

- Mead + Papanicolaou formulated this; implemented in notebook.

$$f(x) = e^{-\sum_{j=0}^{N} \lambda_j x^j} \; ; \text{ define } Z = e^{\lambda_0} = \int_0^1 dx\, e^{-\sum_{j=1}^{N} \lambda_j x^j}$$

Introduce $\Gamma(\lambda_1, \ldots, \lambda_N) = \log Z + \sum_{j=1}^{N} \mu_j \lambda_j$ ⇒ convex; stationary points are solutions of the moment equations.

Bayesian optimization: notebook Bayesian optimization is self-contained.
One application of Bayesian methods for Machine Learning (ML), Bayesian neural network next!

11/13/19

- Background remarks on the role of Bayesian methods in Machine Learning (ML).
  - ML encompasses a broad array of techniques, many of which do not require a Bayesian approach, or even have a philosophy largely counter to the Bayes way of doing things.
  - But there are clearly places where Bayesian methods are useful.
  - We will touch upon two examples:
    i) Bayesian Optimization
    ii) Bayesian Neural Networks
  - Given time limitations, these will necessarily only be teasers for a more complete treatment.

- Step through the Neal Bayesian Methods for Machine Learning selected slides
  1. These come from 2004, which seems out-of-date, but the underlying ML ideas have been around for a long time. Recent successes have stemmed from refinements of old ideas. (some of which were thought not to work, but just needed implementation tweaks).

  2. Bayesian Approach to ML (or anything).
     - emphasis that the approach is very general
     - Some of the uses in step 4) are often different in ML.

  often ML: - Don't account for uncertainty, optimize to find prediction
  ⇒ but some applications require an assessment of risk (medical) or a non-black-box idea of how conclusion reached (legal)

  3. Distinctive features set up to contrast with black-box ML.

  4. "Learning Machine" approach is one way to view ML.
     - Note that it works best with big data, i.e. a lot of data, in which case we know that Bayesian priors become less important.
     - Conversely, the relevance of a Bayesian approach is greater when data is limited (or expensive to get).

11/13/19

5. Challenges of Specifying Models and Priors

- emphasis is on hierachial models (ie. with hyperparameters) and iterative approach.

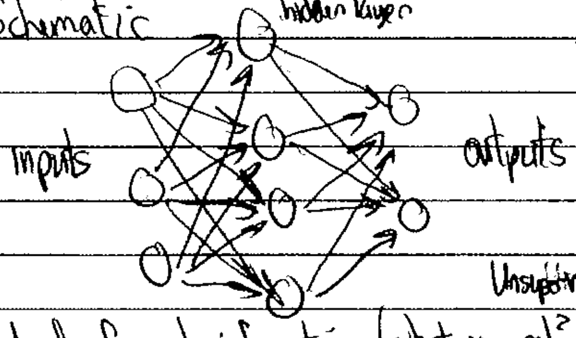
6. Computational Challenge

- We've seen all of these except Variational approximation.
  ⇒ we'll try an example for ML in Mini project IIIb on Bayesian neural networks (BNNs).
- Recall variational methods in physics, which can be powerful and computationally efficient: Given an approximate wavefunction, with parameters controlling its form, an upper bound to the energy is found by take the expectation value of the Hamiltonian with the wavefunction. Adjusting parameters to lower the energy also gives a better wavefunction.
- Now replace the wavefunction by the target posterior.

7. Multilayer Perceptron Neural Networks

- There are various types of neural networks in use, with different strengths. They vary in connectivity, whether signals feed forward only or feed back, etc.
- Schematic

very versatile way to reproduce functions ⇒

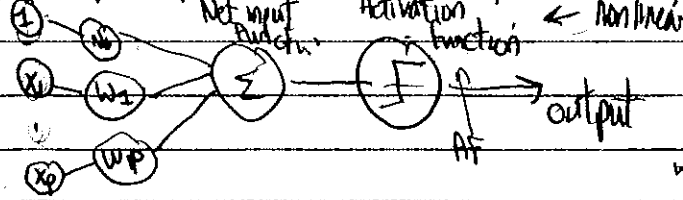


Train with a set of inputs and outputs (supervised learning), which determines the weights that dictate how inputs are mapped to outputs.

Unsupervised learning has inputs and cost function to be minimized.

- Used for classification (what numeral? what phase?) and regression (learn a function).
- Inputs $x_1, \dots, x_p$ depend in detail of the problem
- One node:

$$\overset{AF}{\downarrow}$$
$$y = f\Big(\sum_{j=1}^{p} w_j x_j + \underset{\overset{\uparrow}{\text{or } w_0}}{b}\Big) \Rightarrow$$



in the example one node would be one j. Then $w_{ij} \Rightarrow w_i$, tanh is the AF, and we only consider one output.