Bayesian modelling

Topics in Cognitive Modelling
Jan. 22, 2016

John Lee, Chris Lucas
School of Informatics
University of Edinburgh
{jlee,clucas2}@inf.ed.ac.uk

Recap

Bias-variance tradeoff:

- Learners with **fewer** constraints can learn more things, but have higher variance: they are more sensitive to noise, can overfit, and require more data to generalize correctly.
- Learners with **more** constraints have higher bias: they are more robust to noise and can learn from less data, but may generalize incorrectly if constraints do not match the data.
- The best learner has a high bias that matches the data.

Cognitive science question: what is that bias?

- Another way of asking about domain-specific versus domain-general constraints.

Geman, Bienenstock, & Doursat, 1992

Implicit vs. explicit constraints

The constraints imposed by ANNs are implicit.

- Different architectures can learn different kinds of things.
- In many cases it's hard to quantify the relationship between the architecture and what can be learned.

If we want to study human learning biases, maybe we should be explicit about modelling them.

- This is (part of) the philosophy of the Bayesian approach to cognitive modelling.

Bayesian modelling

Focuses on computational-level questions:

- What is the information available to the learner?
- What is the high-level (mathematical) description of the problem being solved?
- Different algorithms could be used to solve the problem; often no commitment to one or another.

Frames cognition as optimization under uncertainty:

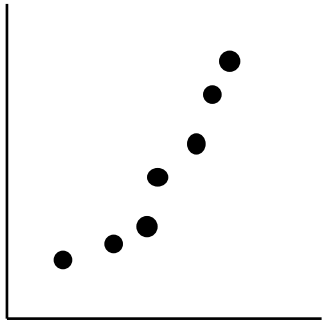
- Choose the best hypothesis or decision using the rules of probability theory.
- Models specify mathematically what the learner's biases are, how observed data affects beliefs about the best hypothesis.

Example: Function learning

Suppose we are trying to predict some response *y* given an input *x*.

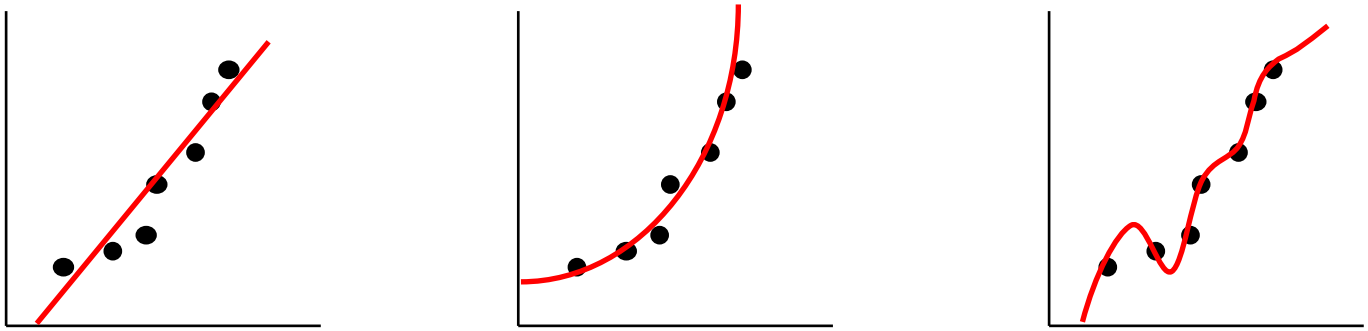
- If I push with *x* force, how far (*y*) does an object move?
- If I add *x* grams of salt, how good (*y*) does my food taste?

We observe (*x,y*) pairs, and we want to learn a function to predict *y* from new *x* (i.e., regression).



Example: Function learning:

Which function is right?



Bayesian function learning

Restating the problem in Bayesian terms:

- The data *d* are the set of observed (*x*,*y*) pairs.
- Each possible function is a hypothesis *h*, and we want to know the probability that any particular hypothesis is correct, given the data we saw.
- The hypothesis space *H* is the set of functions under consideration.

Bayes' Rule

We can now formulate the problem using Bayes' Rule:

$$P(h \mid d) = \frac{P(d \mid h)P(h)}{P(d)}$$

- P(*h*): prior probability of *h*, before seeing any data.
- P(*d*|*h*): likelihood. How well does *h* explain the data?
- P(*h*|*d*): posterior probability of h after observing *d*.
- P(*d*): evidence. The same for all *h* in *H*, so we can usually ignore it. (Normalizing constant.)

•Bayes' rule tells us how observations should affect beliefs, if information is used optimally.

Two types of constraints

- The hypothesis space *H* states the hard constraints on the learner.
  - *H* = linear functions: learner is constrained to this set, cannot learn anything else.
- The prior states the soft constraints on the learner.
  - All linear functions are equally probable.
  - Positive linear functions are more probable than negative linear functions.
  - Functions with slopes near 1 are more probable.
- Hypotheses with low prior prob. can be learned but need more data to do so than hyps with high prior prob.

Simple example

- Let H = linear functions, and $P(h)$ be uniform.
- Need to make further assumptions to compute $P(d|h)$.

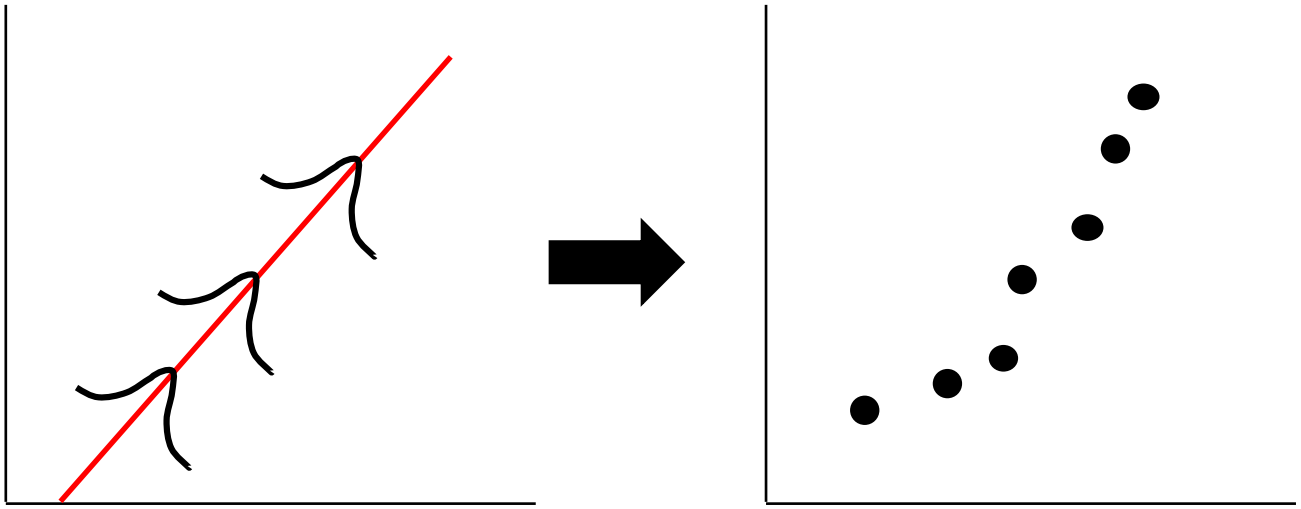
Simple example

- Let *H* = linear functions, and *P(h)* be uniform.
- Need to make further assumptions to compute *P(d|h)*.
  - Data points are generated independently.

$$P(d \mid h) = \prod_i P(d_i \mid h)$$

Simple example

- Let *H* = linear functions, and *P(h)* be uniform.
- Need to make further assumptions to compute *P(d|h)*.
  - Data points are generated independently.
  - Generation process is noisy, produces points distributed around the true function as a Gaussian with variance σ.



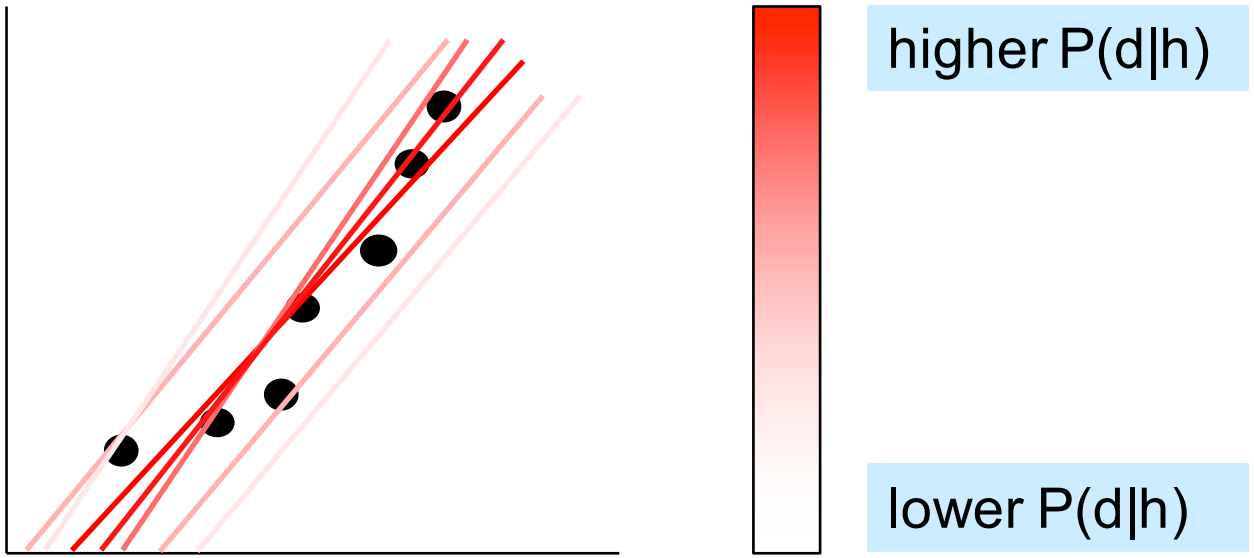
Simple example

- Let *H* = linear functions, and *P(h)* be uniform.
- Need to make further assumptions to compute *P(d|h)*.
  - Data points are generated independently.
  - Generation process is noisy, produces points distributed around the true function as a Gaussian with variance σ.

$$P(d_i \mid h) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(t_i - d_i)^2}{2\sigma^2}\right)$$

Likelihood

- Can now compute *P(d|h)* for any possible line *h*.
  - The *h* with highest *P(d|h)* is called the maximum-likelihood solution; it is the best explanation of the data.



See Goldwater (2010), Griffiths, et al., (2009), Griffiths and Yuille (2006) for further details and examples related to this and the rest of the lecture.

Posterior

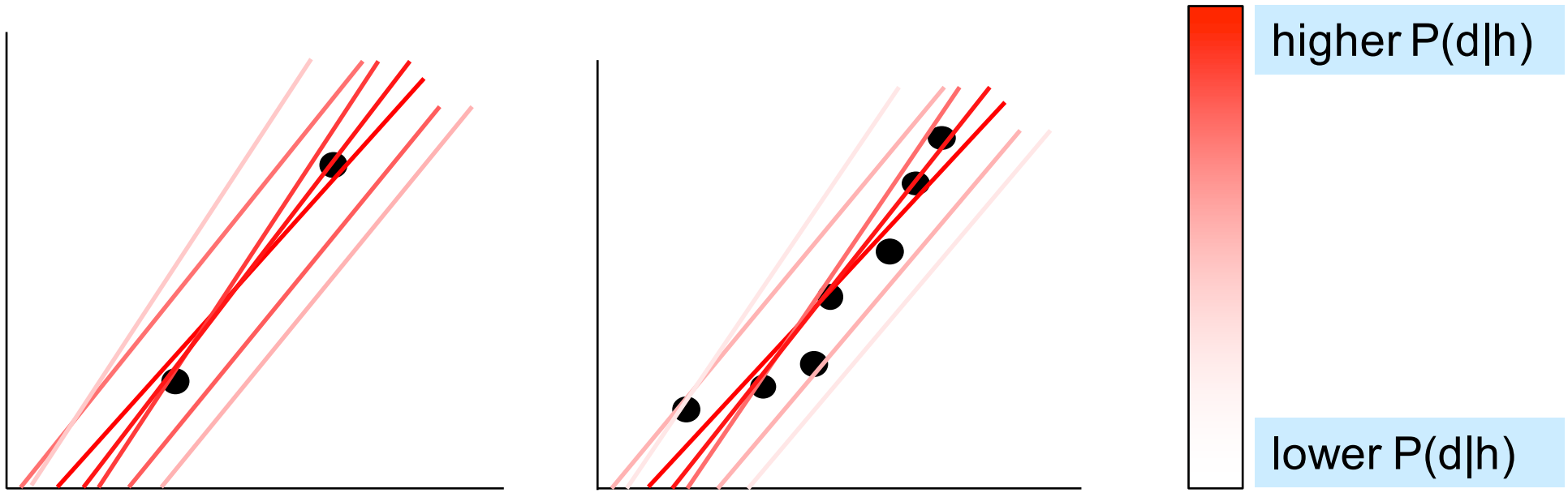
- In this case, since *P(h)* is uniform, the same *h* will also have the highest posterior probability *P(h|d).*

$$P(h \mid d) = \frac{P(d \mid h)P(h)}{P(d)}$$

- With non-uniform prior, *P(h|d)* may differ from *P(d|h).*
  - *P(h|d)* takes into account both how well *h* explains the data, and prior beliefs about *h* (either learning biases, or beliefs obtained from previous experience).

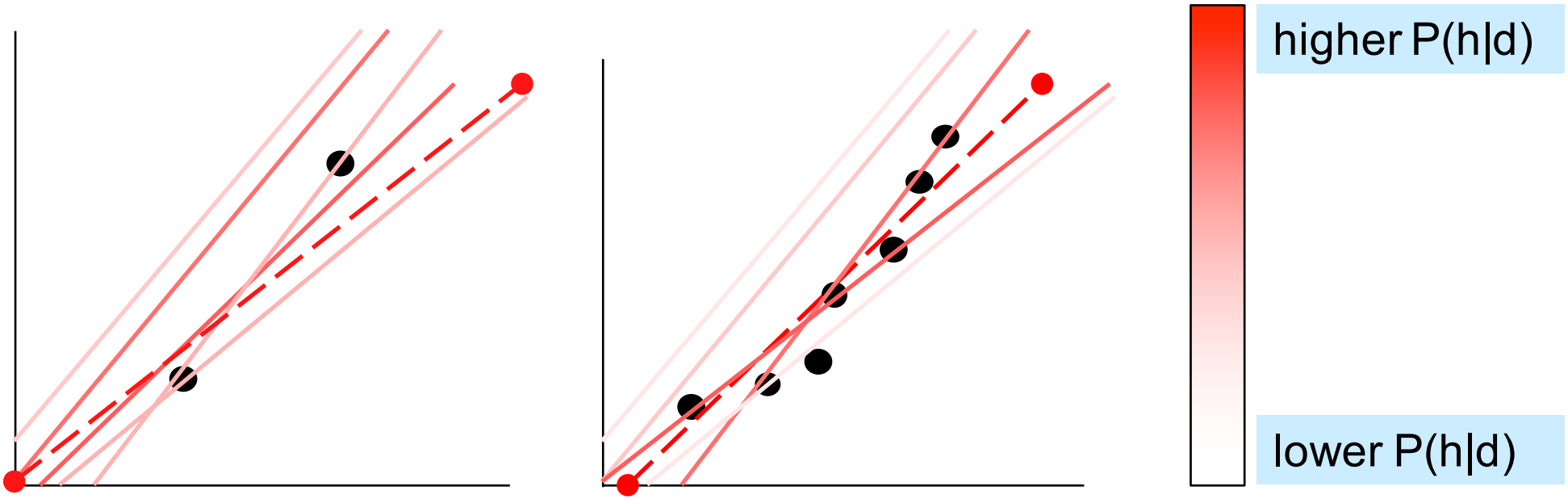
Effect of data

- In general, as more data is observed, a smaller set of hypotheses will have relatively high likelihood.
  - A few points with bad fit vs. lots of points with bad fit.



Effect of data

- Prior isn't affected by data, so as more data accumulates, likelihood becomes more important.
  - Imagine prior prefers lines passing nearer the origin.



What can the model tell us?

- Model made assumptions about the hypothesis space, priors, and how the data were generated.
  - In a cognitive model, we assume these are the constraints built in to the human learner*.
- Under these conditions, what predictions are made (e.g. the *y* value for a new *x* point)?
- Do these predictions match those of humans?
  - If so, suggests humans have similar constraints, and make probabilistically optimal predictions (like the model).
  - If not, either the constraints are wrong, or humans are not optimal. Further investigation may help decide.

What can't the model tell us?

- This model defines the problem being solved and the optimal solution, but not *how* to find the solution.
- Lots of solution methods possible:
  - Analytical: use calculus to derive the answer.
  - Algorithmic: use gradient descent or some other iterative procedure (e.g., Markov chain Monte Carlo)
  - Guess and check?
- Even if humans behave as the model does, we don't know how they found the solution.

Hypothesis averaging

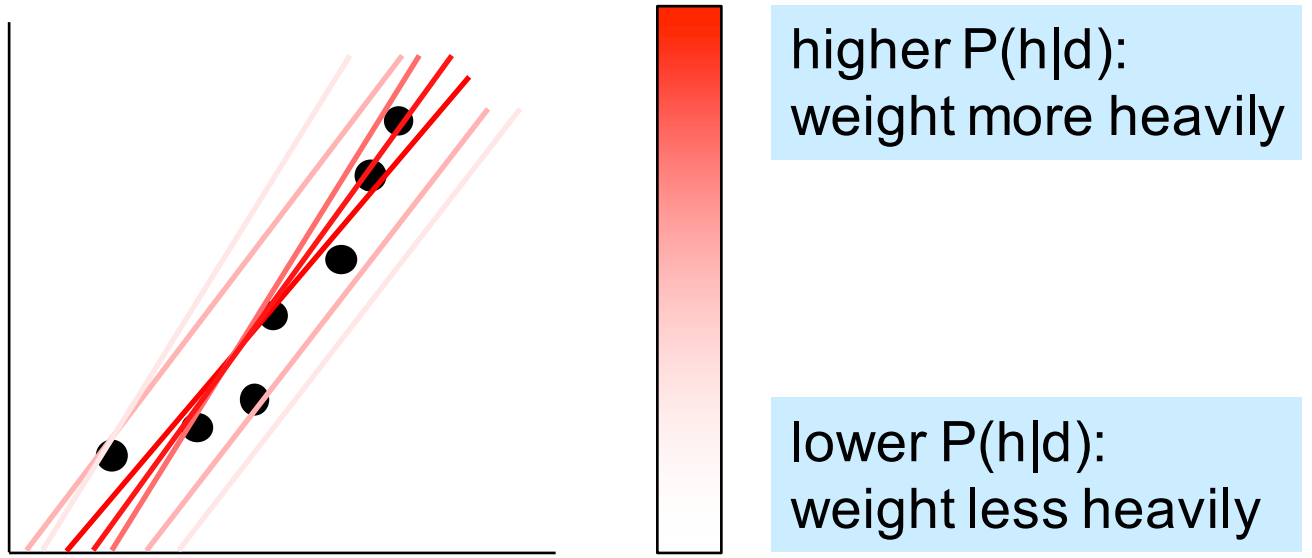
- Another important feature of many Bayesian models.
  - When making a prediction, do not base it on the single best hypothesis (highest *P(h|d)*); instead average over all possible hypotheses, weighted by their probabilities.

$$P(d_{n+1} \mid d_1 \ldots d_n) = \int_{h \in H} P(d_{n+1} \mid h) P(h \mid d_1 \ldots d_n)$$

Replace with Σ if *H* is discrete.

Hypothesis averaging

- Another important feature of many Bayesian models.
  - When making a prediction, do not base it on the single best hypothesis (highest $P(h|d)$); instead average over all possible hypotheses, weighted by their probabilities.



Representation

- Like ANNs, Bayesian models can use various types of representations.
  - Symbolic (localized): different discrete symbol for each input.
  - Feature-based (distributed): feature vectors represent each input, as in many ANNs.
- Symbolic representations may be viewed as convenient shorthand rather than mentally real.
- Choice of representation can have a big effect (as in ANNs).

Relationship to other approaches

- Bayesian approach is compatible with both nativist and empiricist views.
  - Depends on whether hypothesis space and priors are domain-specific or domain-general.
- Emphasis on making explicit both constraints and effects of data on beliefs (i.e., learning).
- Little emphasis on how the probabilistic computations might be carried out in the brain.
  - Maybe using something like an ANN?
  - May use symbolic or distributed representations.

Conclusion

ANNs define the learner's behaviour algorithmically:

- Compute functions from input data to output vector.
- Trained from input-output pairs (supervised) or just input data (unsupervised).
- Architecture implicitly defines their learning biases.

Bayesian models define the learner's behavior mathematically.

- Define functions from input data to output distribution, assume optimal behaviour.
- Also can be supervised or unsupervised, using various training algorithms.
- Hypothesis space and prior explicitly defines their learning biases.

Reminders

- Responses to your choice of segmentation modelling paper due in class on Tuesday (details on website).

References

Geman, S., Bienenstock, E., & Doursat, R. 1992. Neural networks and the bias/variance dilemma. *Neural computation*, *4*(1), 1-58.

Goldwater, Sharon. 2010. Bayesian Modelling. Notes from Lecture 12 of Computational Cognitive Science, Autumn 2010.

Griffiths, T. L., Lucas, C. G., Williams, J. J., & Kalish, M. L. 2009. Modeling human function learning with Gaussian processes. *Advances in Neural Information Processing Systems 21.*

Griffiths, Tom L. and Alan Yuille. 2006. A primer on probabilistic inference. *Trends in Cognitive Sciences* 10(7). Supplement to special issue on Probabilistic Models of Cognition.