Lecture 26

WRAPUP

Basic skills you will take away

- probabilistic thinking
- how machine learning works
- disciplined probabilistic modeling
- computation over math: sampling
- you must think of inference, not just point estimates
- Dont Overfit

Key skill you now have

- you might not feel entirely confident, but...
- you know how to come up with a *generative* story for your data..
- and model it, however imperfectly...
- iterating on your model..
- while checking and testing it..

Bayesian

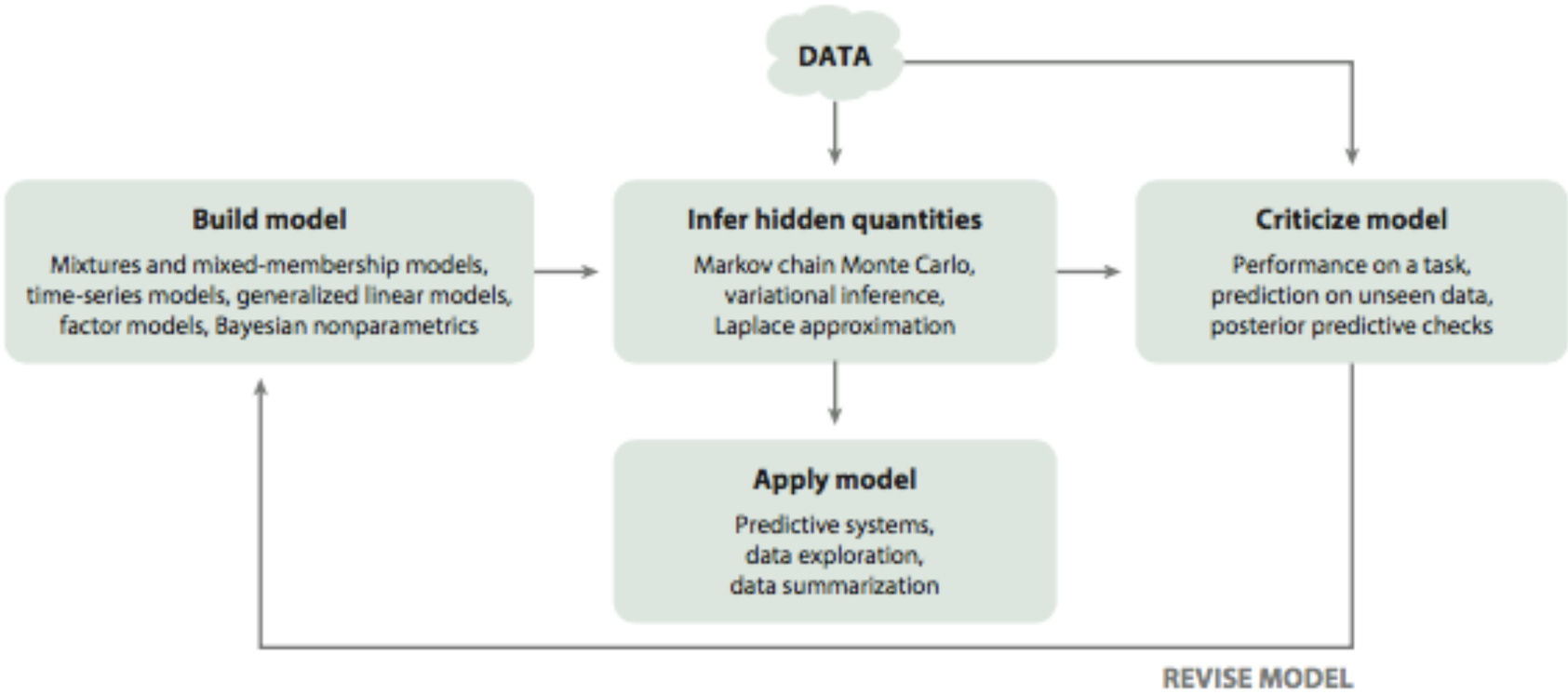
- sample is the data, and is fixed
- parameter is stochastic, has prior and posterior distribution
- posterior: $p(\theta|y) = \frac{p(y|\theta)\,p(\theta)}{p(y)}$, can summarize via MAP
- just bayes rule: $posterior = \frac{likelihood \times prior}{evidence}$

From `edwardlib`: $p(\mathbf{x} \mid \mathbf{z})$

describes how any data $\mathbf{x}$ depend on the latent variables $\mathbf{z}$.

- **The likelihood posits a data generating process**, where the data $\mathbf{x}$ are assumed drawn from the likelihood conditioned on a particular hidden pattern described by $\mathbf{z}$.
- The *prior* $p(\mathbf{z})$ is a probability distribution that describes the latent variables present in the data. **The prior posits a generating process of the hidden structure**.

Box's loop


DATA
Build model
Mixtures and mixed-membership models, time-series models, generalized linear models, factor models, Bayesian nonparametrics
Infer hidden quantities
Markov chain Monte Carlo, variational inference, Laplace approximation
Criticize model
Performance on a task, prediction on unseen data, posterior predictive checks
Apply model
Predictive systems, data exploration, data summarization
REVISE MODEL

What did we cover?

- Basic Stats and sampling
- The nature of learning and machine learning
- stochastic optimization with sgd and simulated annealing
- MCMC and latent variables (parameters or otherwise)
- slice sampling, hmc and NUTS
- hierarchical models
- glms

- problems with samplers
- model checking
- model comparison and ensembling
- decision Theory
- mixture models
- supervised vs unsupervised vs semisupervised learning
- EM and VI including ADVI
- Gaussian Processes

Who uses this?

- Machine Learning: either Bayesian or Discriminant
- Recommendations, Ranking, etc mixed effects models
- See StitchFix Algorithms tool
- Genetics, Ecology. Psychology, Astronomy, basically, everyone
- NLP: LDA, Naive Bayes, etc
- hyperparameter optimization using GPs
- understanding inference in NNs. See http://mlg.eng.cam.ac.uk/yarin/blog_3d801aa532c1ce.html

A particular example: Bayesian Bandits

See textbook chapter 6

Suppose you are faced with NN slot machines (colourfully called multi-armed bandits). Each bandit has an unknown probability of distributing a prize (assume for now the prizes are the same for each bandit, only the probabilities differ). Some bandits are very generous, others not so much. Of course, you don't know what these probabilities are. By only choosing one bandit per round, our task is devise a strategy to maximize our winnings.

- Ted Dunning

- Find the best bandit, and as quickly as possible. Complicated by "best" being a probability distribution
- Reinforcement Learning exploration vs. exploitation dilemma: Suppose we found a good-enough bandit. Do we keep using it, or try and find another?
- Examples: Internet display advertising: A/B/C/D testing strategies); Ecology: what foraging strategy should an animal use? Whats out investment strategy?
- Need to formulate decision making under a loss calculated from a posterior

From Cam-Davidson Pilon:

The Bayesian solution begins by assuming priors on the probability of winning for each bandit. So a very natural prior is the flat prior over 0 to 1. The algorithm proceeds as follows:

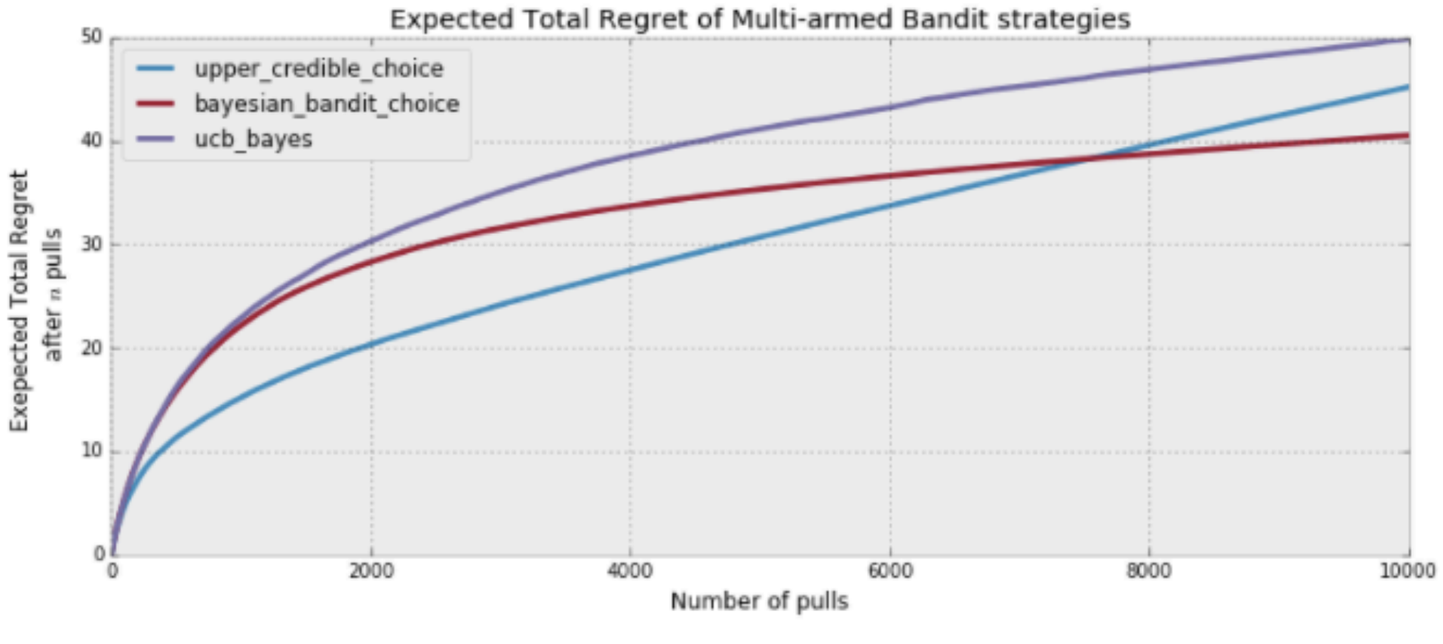
For each round:

1. Sample a random variable X_b from the prior of bandit b, for all b.
2. Select the bandit with largest sample, i.e. select $B = \arg\max_b X_B$
3. Observe the result of pulling bandit B, and update your prior on bandit B.
4. Return to 1.

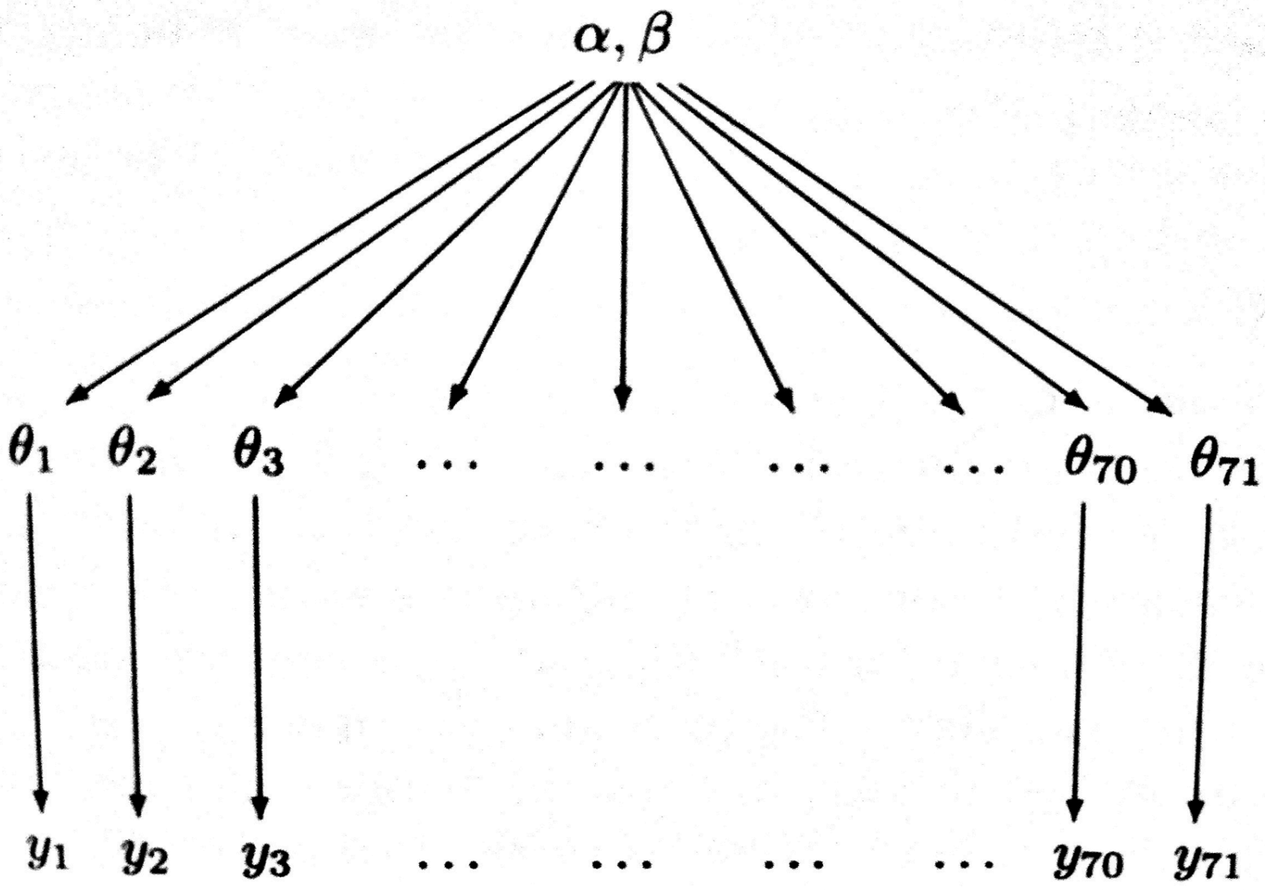
   Since the initial priors are Beta(α=1,β=1) and the observed result X (a win or loss, encoded 1 and 0 respectfully) is Binomial, the posterior is a Beta(α=1+X,β=1+1−X).

Regret Measure (Risk)

$$\bar{R}_T = E[\sum_{i=1}^{T}(w_{opt} - w_B(i))]$$



Hierarchical Models



Key Idea: Share statistical strength

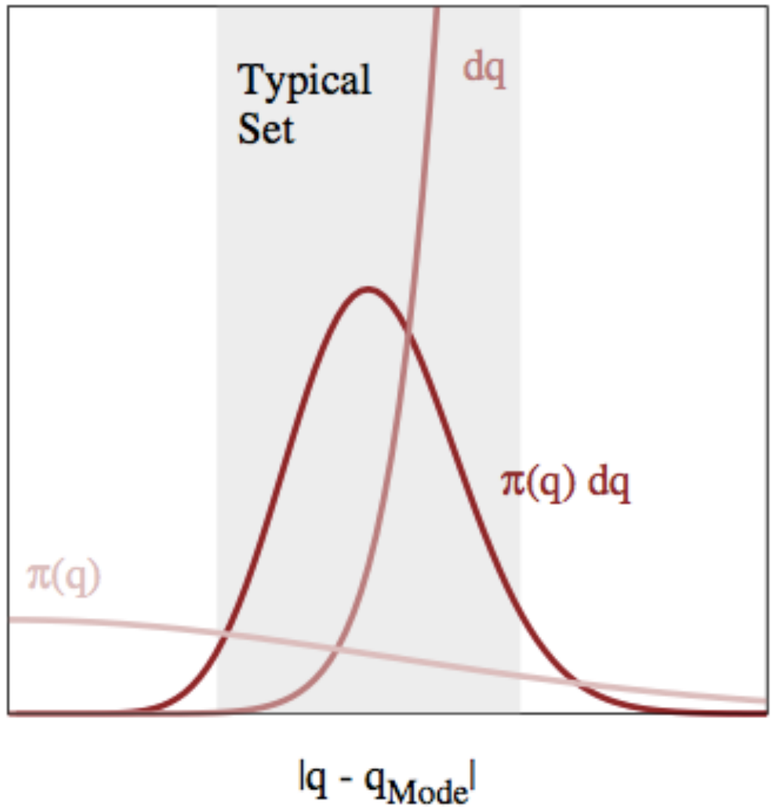
- Some **units** (experiments) statistically more robust
- Non-robust experiments have smaller samples or outlier like behavior
- Borrow strength from all the data as a whole through the estimation of the hyperparameters
- **regularized partial pooling model** in which the "lower" parameters (θs) tied together by "upper level" hyperparameters.

Fixing And Speeding up Samplers

- first check traces and autocorrelation
- see if burnin and thinning help
- see Gewecke
- get 2 chains if possible (scale your problem down) and calculate n_{eff} and Gelman-Rubin
- do pairwise correlation SPLOMs to see if parameters are correlated.
- Decorrelate by centering

- marginalize over discretes by hand
- check energy matching plots to see if you are having good coverage in HMC
- In Hierarchicals, check for divergences, decrease step size to see if they go away
- if funnel is too curved, divergences may persist and energy plots will be bad
- try to de-hierarchicalize by doing non-centered decoupling
- posterior predictive checks, always. Especially counterfactual ones.

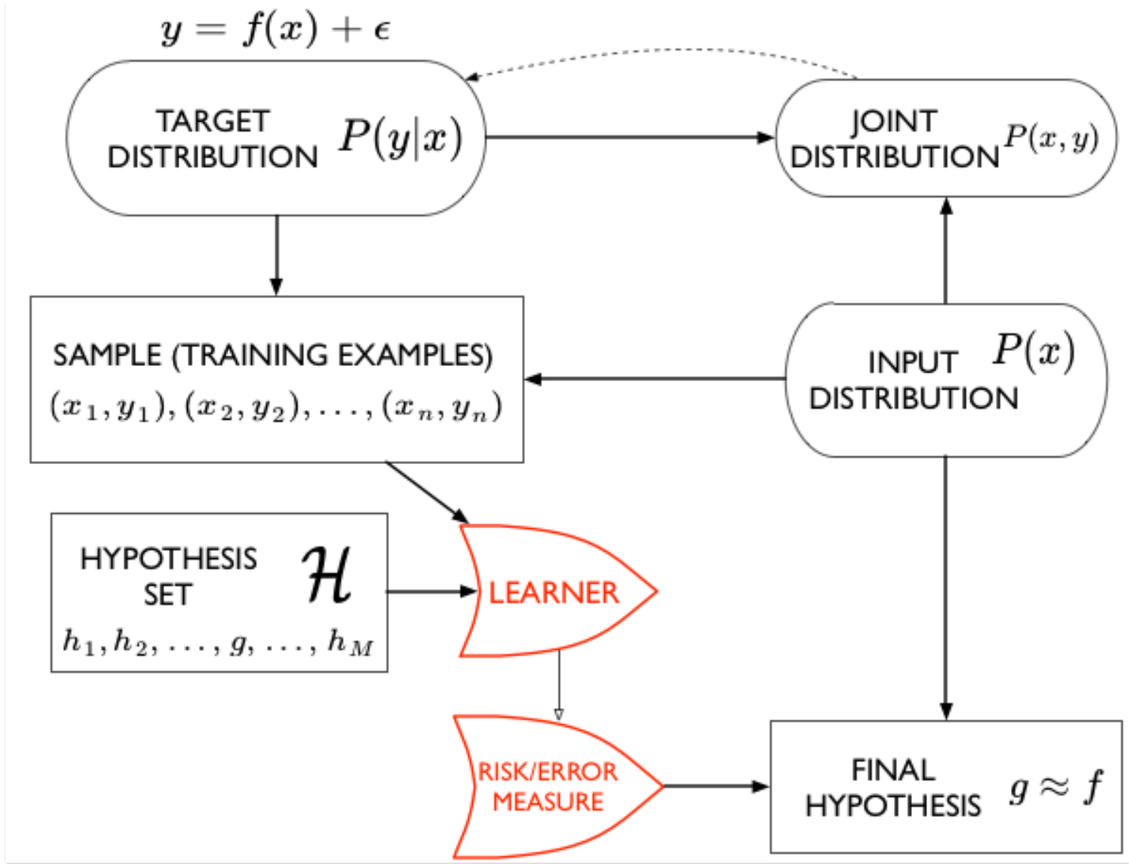
Critical: explore the typical set



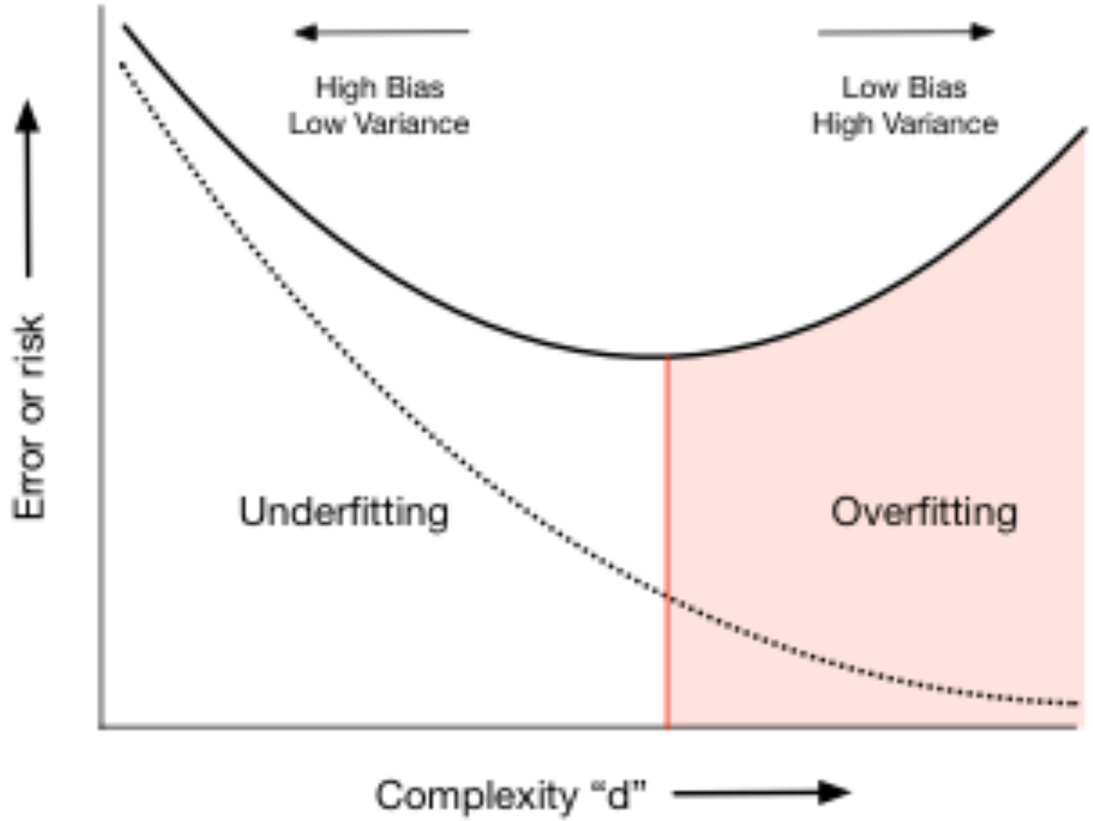
Different kinds of learning

- generative in the sense of having a story
- discriminative vs generative in the $p(z|x)$ pr $p(x|z)$ sense
- bayesian vs discriminant in the probabilistic model vs ERM sense.
- supervised (zs known) vs unsupervised (zs not known, likelihood has log of sums)
- semi-supervised (likelihood has $p(x|\theta)$ from "validation" set)

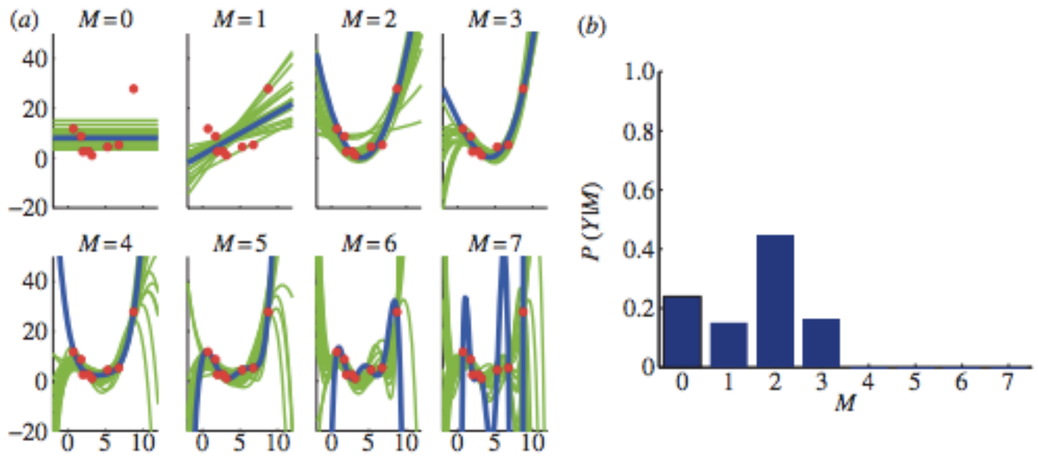
The nature of learning

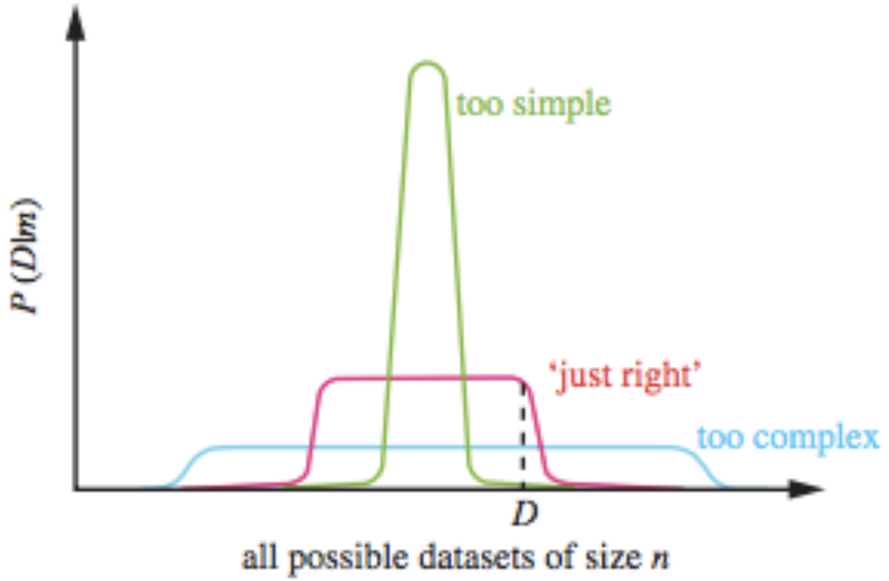


Dont Overfit



Occams Razor





Notes I owe you

- NUTS
- why not reject outside a distribution's support
- varying slopes and intercepts model
- fighting dispersion with continuous mixture models
- parallel tempering

Questions

- Some pointers to scaling these methods will be useful. A few weeks back I tried a simple hierarchical model on a data set at work. This was some customer data with ZIP code information. I was attempting to model individual -> zip code -> county -> state -> country hierarchy and quickly realized it would be too humongous to run on my laptop.
- From the theory it seemed the ability to do online learning was natural for Bayesian models. But in practice, when you get more data (observations), wouldn't it be necessary to stick it into the likelihood (along with all the earlier observations) and do sampling all over again?
- When you suspect there is some hierarchical structure in the data, is there some principled way to "discover" a hierarchical model structure (assuming we don't have a domain expert who pulls a structure out of her hat)? I can think of doing hierarchical clustering first, but I guess that might miss out on some useful details as it considers individual data points, not their distributions.

Where to go from here?

Immediate Tip: Use Stan manual and examples for compendium of models, even if you dont want to be bayesian. Learn pystan, and go to the Stan meetups.

For priors, see: Stan's prior recommendations

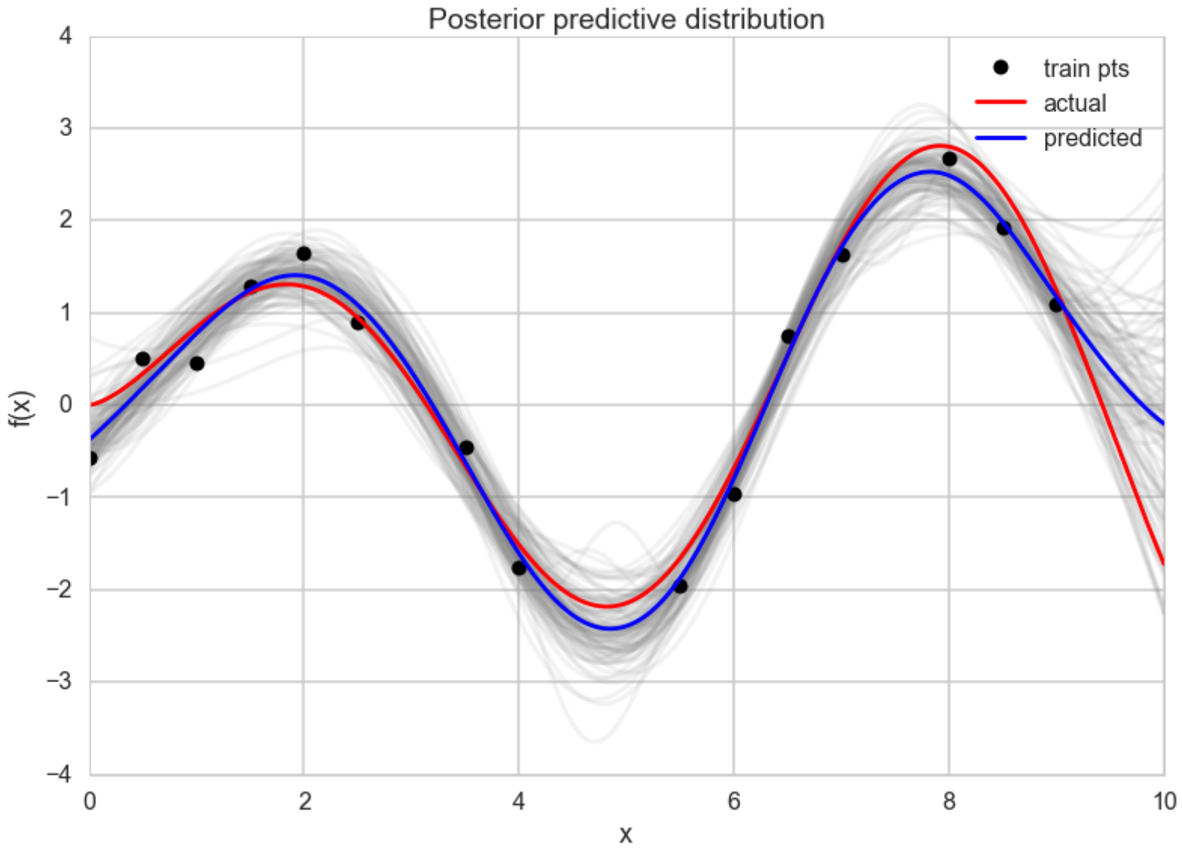
Skim Gelman. Our major topics correspond fairly cleanly to chapters in there.

Work through McElreath and/or the puppies book (Kruschke)

What to learn next?

- Worth your time: Learning Theory and Machine Learning
- Skim: Probabilistic Graphical Models
- Nando De-Freitas Machine Learning
- CS281 next semester. You are well prepared for this. If remote or self-motivated, follow the readings and videos here
- glm's and decision theory course are worth your while (but were offered this semester)

INFERENCE: Posterior (predictive) curves



GOODBYE

- Keep in Touch. I'm around, always there to talk and help!
- Congratulations to those graduating and in new jobs!
- The site will always be there. Will be refactored and improved over summer. Will add missing lectures on HMMs, Kalman Filters.
- please suggest changes