Lecture 26

END OF DAYS

What was this course about?

- **Density Estimation**. (Also called unsupervised or representation learning)
- **Generative Models** in statistics and machine learning..a principled way of modeling (both supervised and unsupervised)
- **Being Bayesian**: a self-consistent process to carry out this modeling
- **Sampling and stochastic optimization**: the technology needed

Along the way we

- learn how to regularize models
- deal with data computationally large/small and statistically small/large
- learn how to optimize objective functions such as loss functions using Stochastic Gradient Descent
- Perform sampling and MCMC to solve a variety of problems, especially Bayes
- Learn how to use parametric, and non-parametric methods

Concepts running through:

Hidden Variables, marginalized

Testing, testing, testing

Differentiation vs Integration

Frequentist vs Bayesian

Generative Models

SMALL WORLD vs BIG WORLD

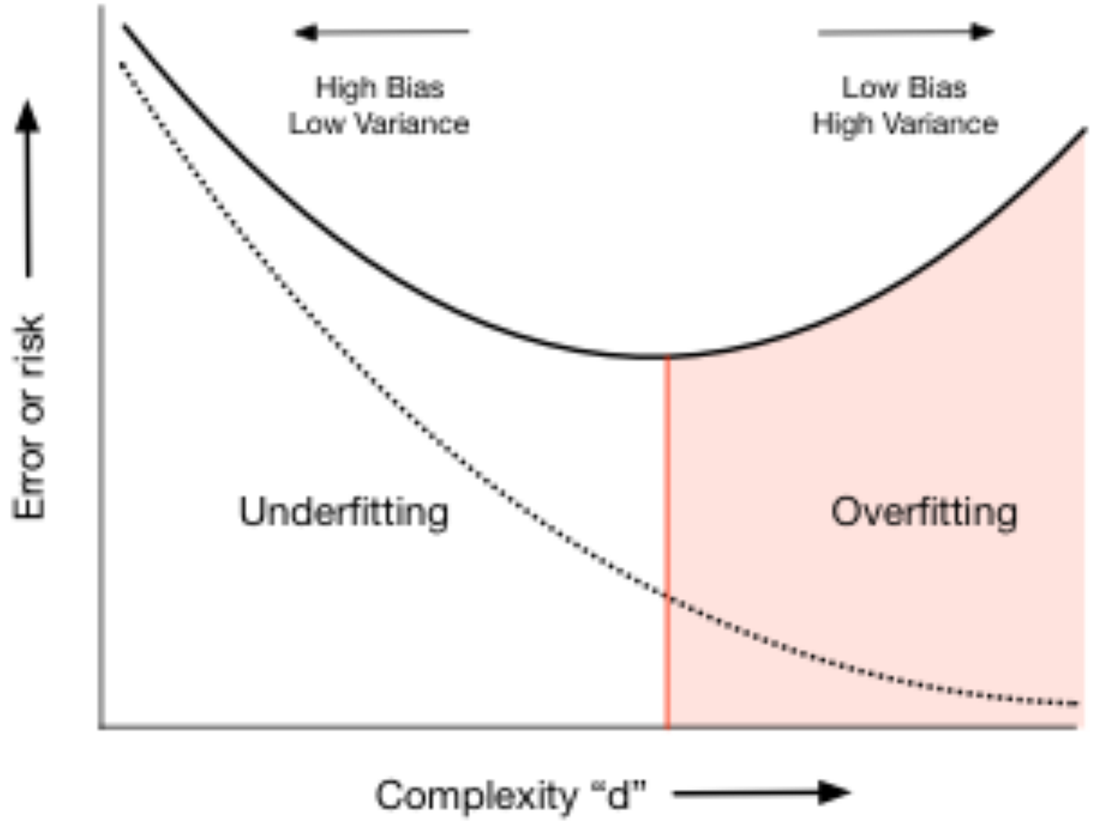
Small world:

$$P(\theta \mid D) = \frac{P(D \mid \theta) \times P(\theta)}{P(D)}$$

Big World:

$$P(M \mid D) = \frac{P(D \mid M) \times P(M)}{P(D)}$$

Dont Overfit



KL-Divergence: compare model to nature

$$D_{KL}(p,q) = E_p[log(p) - log(q)] = E_p[log(p/q)]$$
$$= \sum_i p_i log(\frac{p_i}{q_i}) \; or \int dP log(\frac{p}{q})$$

$$D_{KL}(p,p) = 0$$

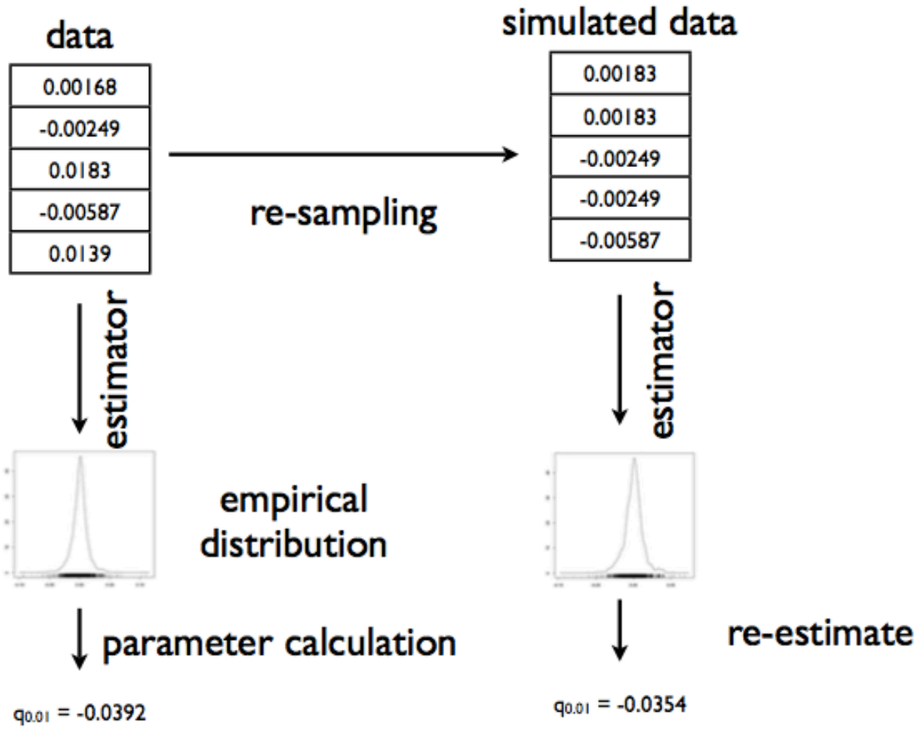
KL divergence measures distance/dissimilarity of the two distributions p(x) and q(x).

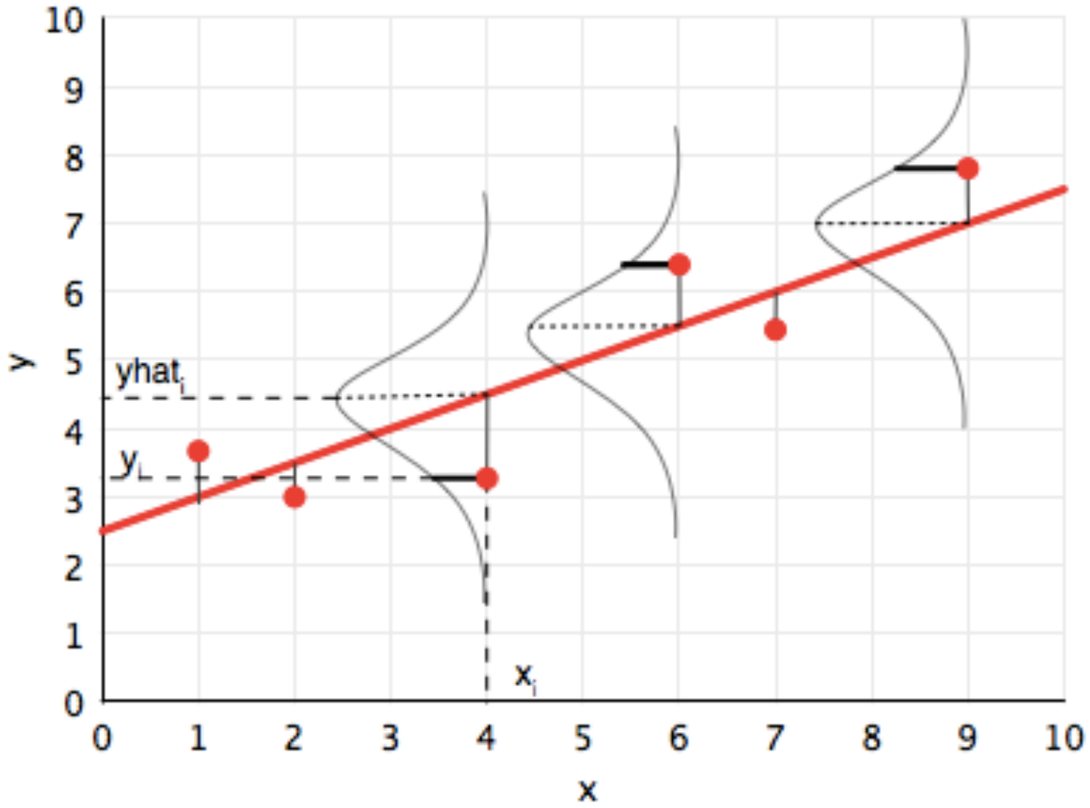
- used for VI, EM, a probabilistic loss function

Frequentist Statistics

"data is a **sample** from an existing **population**"

- data is stochastic, variable; parameters fixed
- fit a parameter
- samples (or bootstrap) induce a sampling distribution on any estimator
- example of a very useful estimator: MLE


data
0.00168
-0.00249
0.0183
-0.00587
0.0139
re-sampling
simulated data
0.00183
0.00183
-0.00249
-0.00249
-0.00587
estimator
estimator
empirical
distribution
parameter calculation
re-estimate
$q_{0.01}$ = -0.0392
$q_{0.01}$ = -0.0354


yhat$_i$
y_i
x_i
x
y

Information Entropy and MAXENT

$$H(p) = -E_p[log(p)] = -\int p(x) log(p(x)) dx \;\; OR \; -\sum_i p_i log(p_i)$$

- what would be the least surprising distribution, the one with the least additional assumptions (most conservative), the one that can happen in the most ways consistent with constraints
- most common distributions used as likelihoods (and priors) are in the exponential family, MAXENT subject to different constraints.

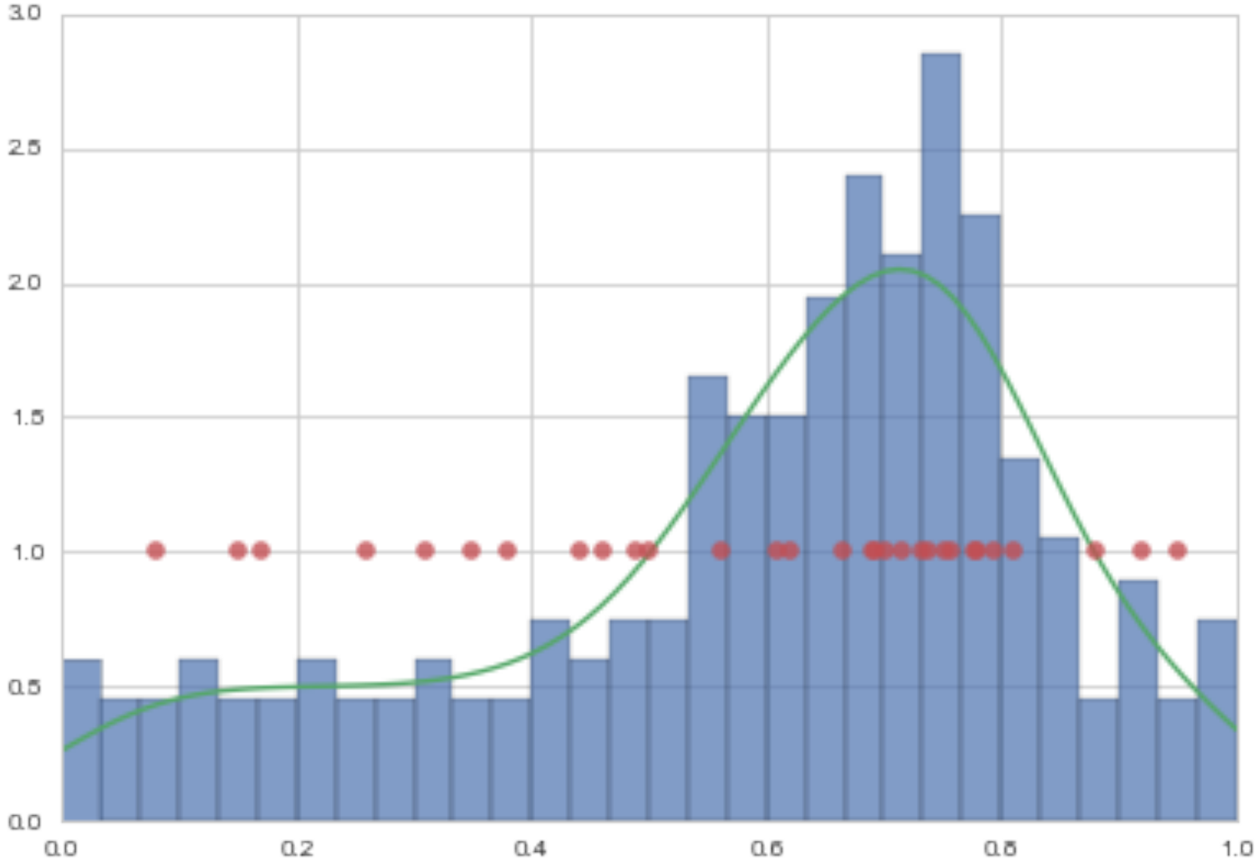
SAMPLE vs POPULATION

Want: $R_{out}(h) = E_{p(x)}[(h(x) - f(x))^2] = \int dx p(x)(h(x) - f(x))^2$

LLN:

$$R_{out}(h) = \lim_{n\to\infty} \frac{1}{n} \sum_{x_i \sim p(x)} (h(x_i) - f(x_i))^2 = \lim_{n\to\infty} \frac{1}{n} \sum_{x_i \sim p(x)} (h(x_i) - y_i)^2$$

$$\mathcal{D} \text{ representative } (\mathcal{D} \sim p(x)) \implies \mathcal{R}_{\mathcal{D}}(h) = \sum_{x_i \in \mathcal{D}} (h(x_i) - y_i)^2$$



Statement of the Learning Problem

The sample must be representative of the population!

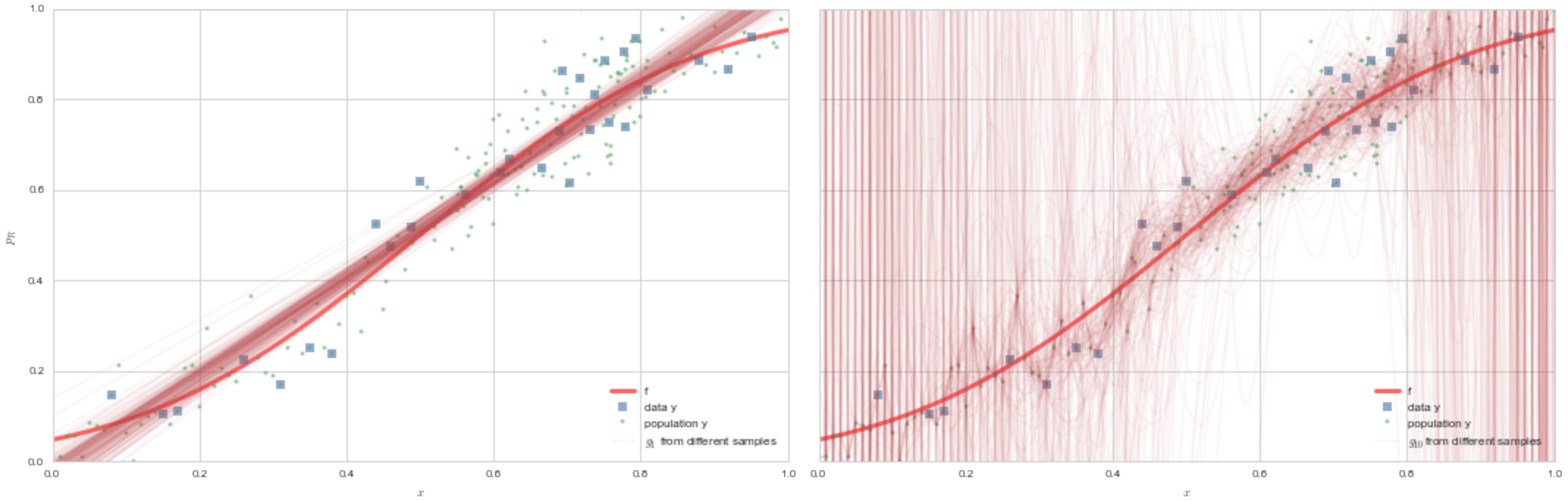
$$A : R_{\mathcal{D}}(g) \ smallest \ on \ \mathcal{H}$$
$$B : R_{out}(g) \approx R_{\mathcal{D}}(g)$$

A: Empirical risk estimates in-sample risk.

B: Thus the out of sample risk is also small.

UNDERFITTING (Bias)

vs OVERFITTING (Variance)



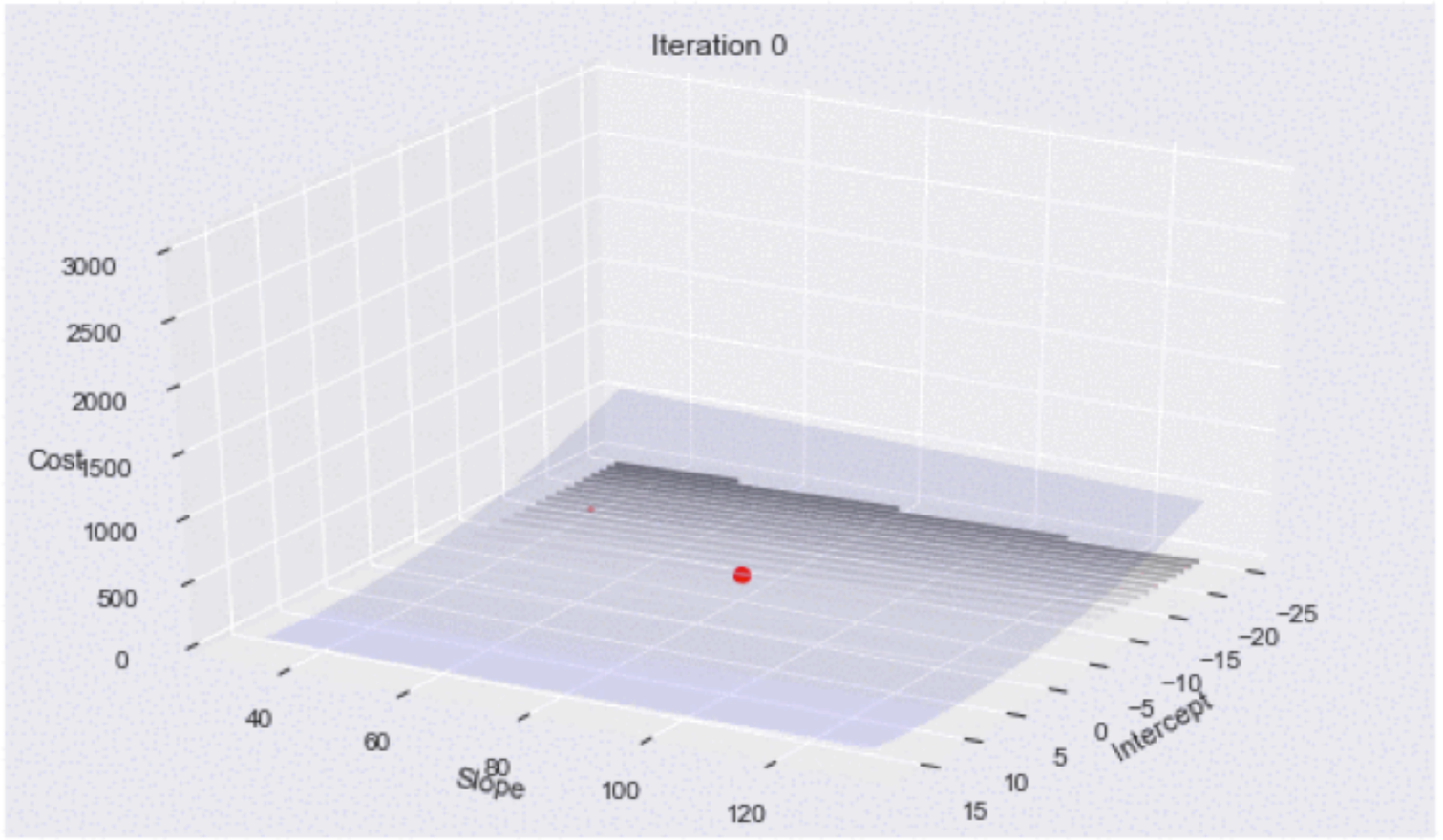
Stochastic Gradient Descent

$$\theta := \theta - \alpha \nabla_\theta J_i(\theta)$$

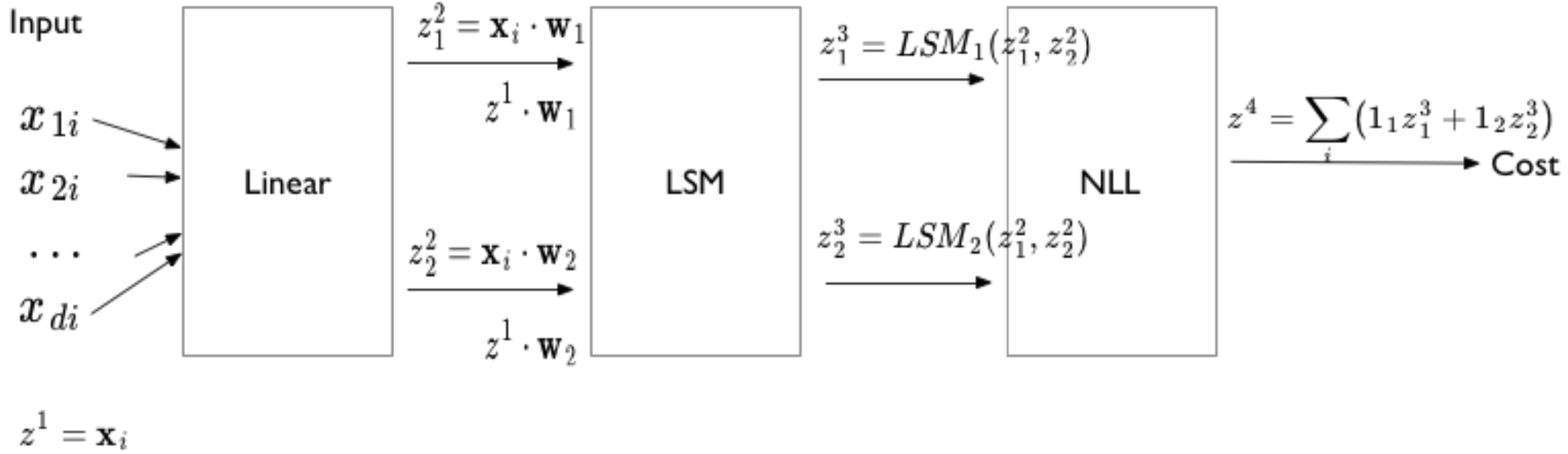
ONE POINT AT A TIME

```
for i in range(nb_epochs):
  np.random.shuffle(data)
  for example in data:
    params_grad = evaluate_gradient(loss_function, example, params)
    params = params - learning_rate * params_grad
```

Mini-Batch: do some at a time


Iteration 0
3000
2500
2000
Cost
1500
1000
500
0
40
60
80
100
120
Slope
15
10
5
0
−5
−10
−15
−20
−25
Intercept

Softmax Formulation of Logistic Regression



Backprop: Reverse Mode Differentiation

$$Cost = f^{Loss}(\mathbf{f^3}(\mathbf{f^2}(\mathbf{f^1}(\mathbf{x}))))$$

$$\nabla_{\mathbf{x}} Cost = \frac{\partial f^{Loss}}{\partial \mathbf{f^3}} \frac{\partial \mathbf{f^3}}{\partial \mathbf{f^2}} \frac{\partial \mathbf{f^2}}{\partial \mathbf{f^1}} \frac{\partial \mathbf{f^1}}{\partial \mathbf{x}}$$

Write as:

$$\nabla_{\mathbf{x}} Cost = (((\frac{\partial f^{Loss}}{\partial \mathbf{f^3}} \frac{\partial \mathbf{f^3}}{\partial \mathbf{f^2}}) \frac{\partial \mathbf{f^2}}{\partial \mathbf{f^1}}) \frac{\partial \mathbf{f^1}}{\partial \mathbf{x}})$$

Law of Large numbers, LOTUS, MC

Let $x_1, x_2, \ldots, x_n$ be a sequence of IID values from random variable X, which has finite mean μ. Let:

$$S_n = \frac{1}{n}\sum_{i=1}^{n} x_i, \text{ then } S_n \to \mu \, as \, n \to \infty.$$

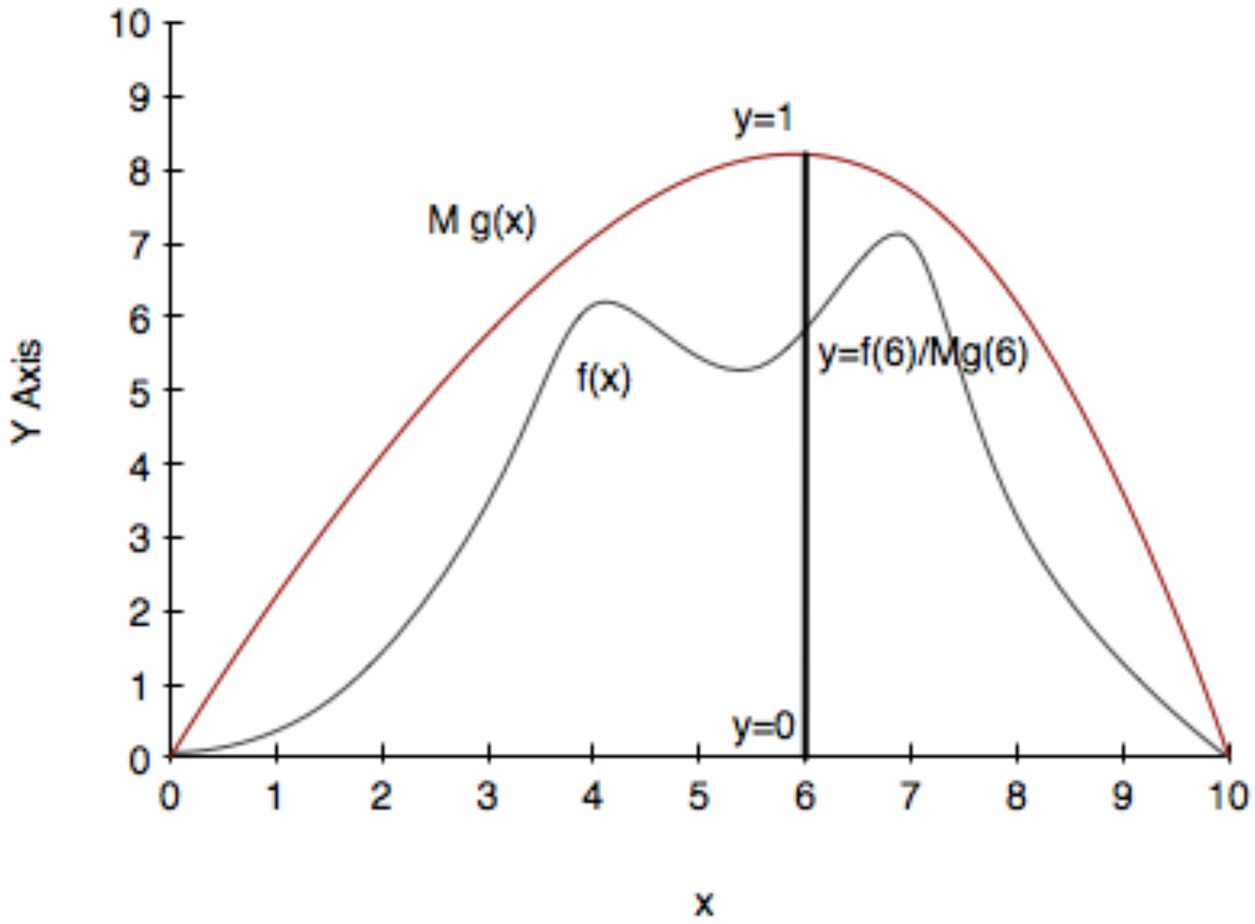
- Expectations become sample averages. Convergence for large N.

$$E_f[g] = \int g(x)dF = \int g(x)f(x)dx = \lim_{n\to\infty} \frac{1}{N}\sum_{x_i \sim f} g(x_i)$$

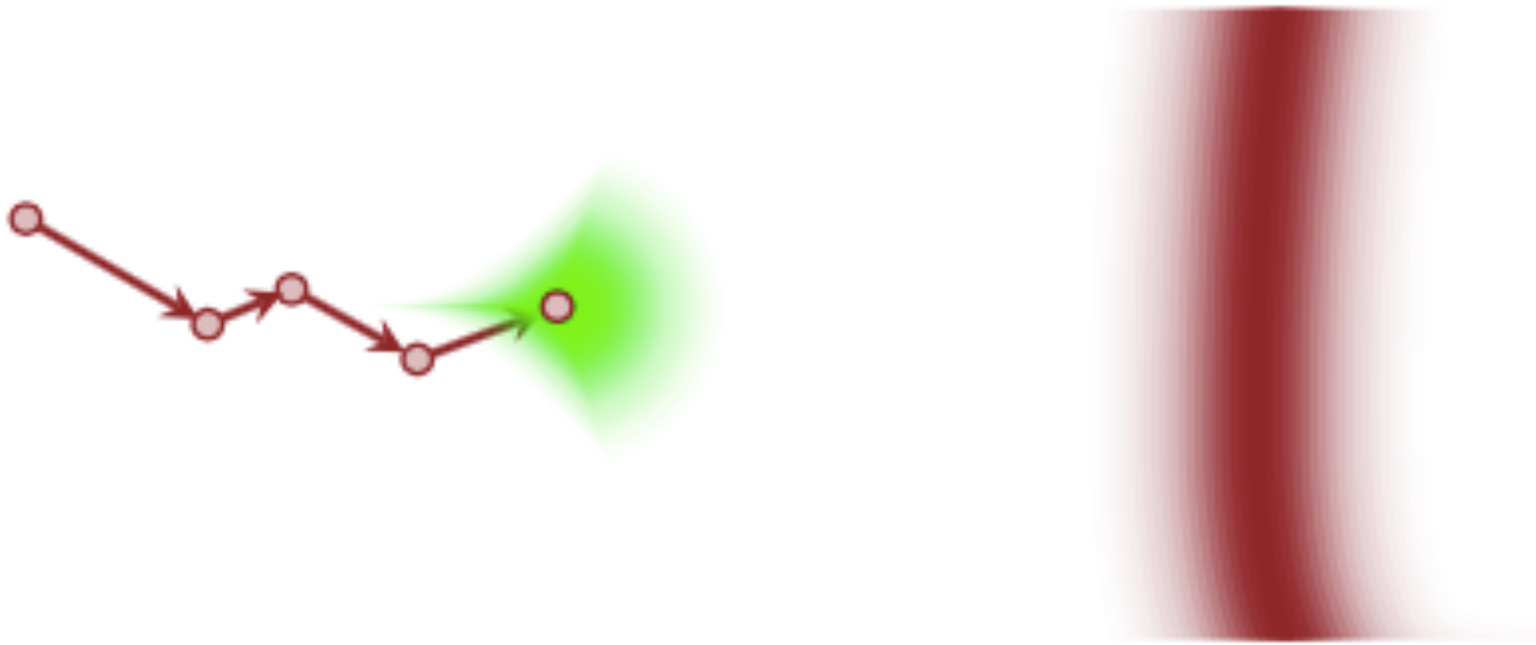
- allows for monte-carlo

NEED SAMPLES: GENERATE THEM!

- Inverse method, Rejection (on steroids)
- Stratification to reduce variance
- Importance (for expectations)
- MCMC, MH, HMC, Slice, ADVI, etc
- integrals (marginalize) by ignoring dimensions in histogram

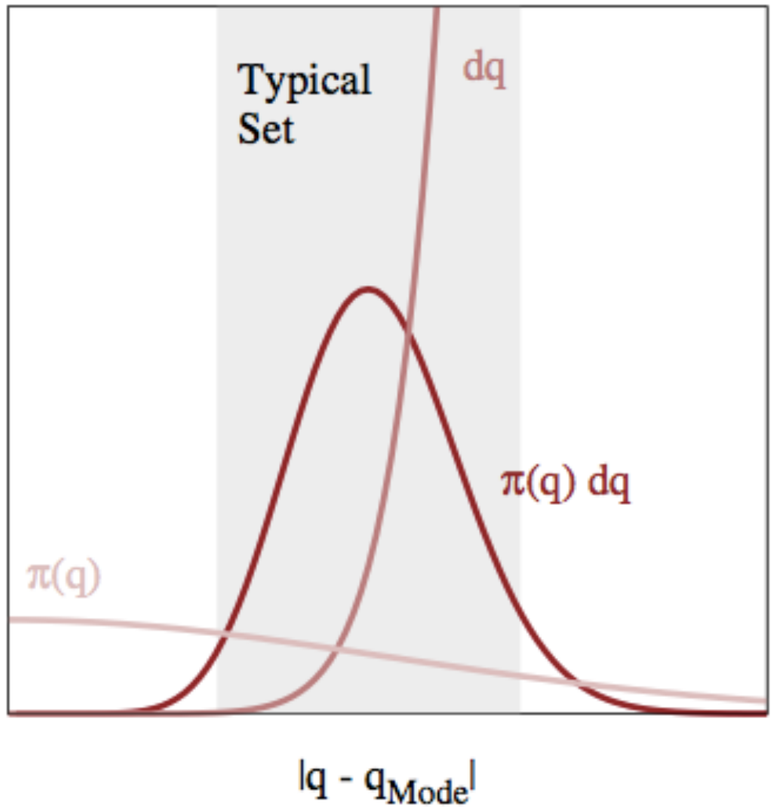


MCMC Intuition: proposal approaches typical set

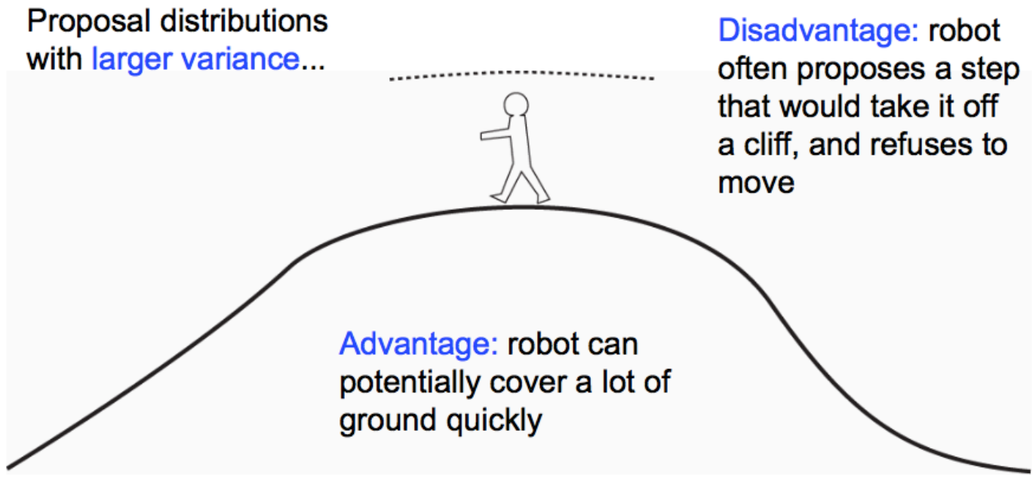


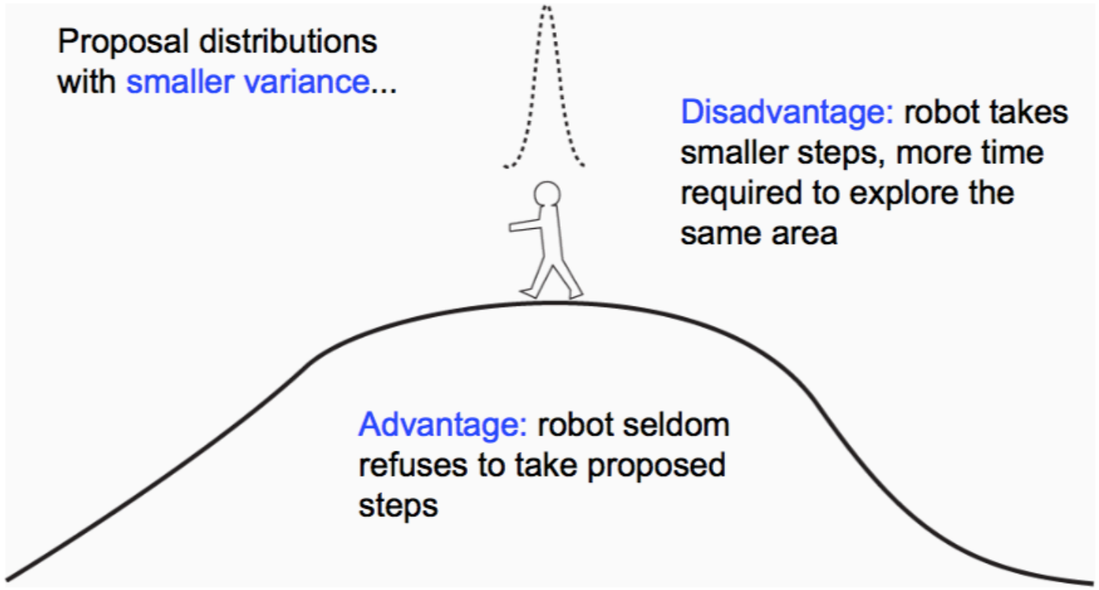
Instead of sampling p we sample q, yielding a new state, and a new proposal distribution from which to sample.

Critical: explore the typical set: stationarity



Metropolis and MH



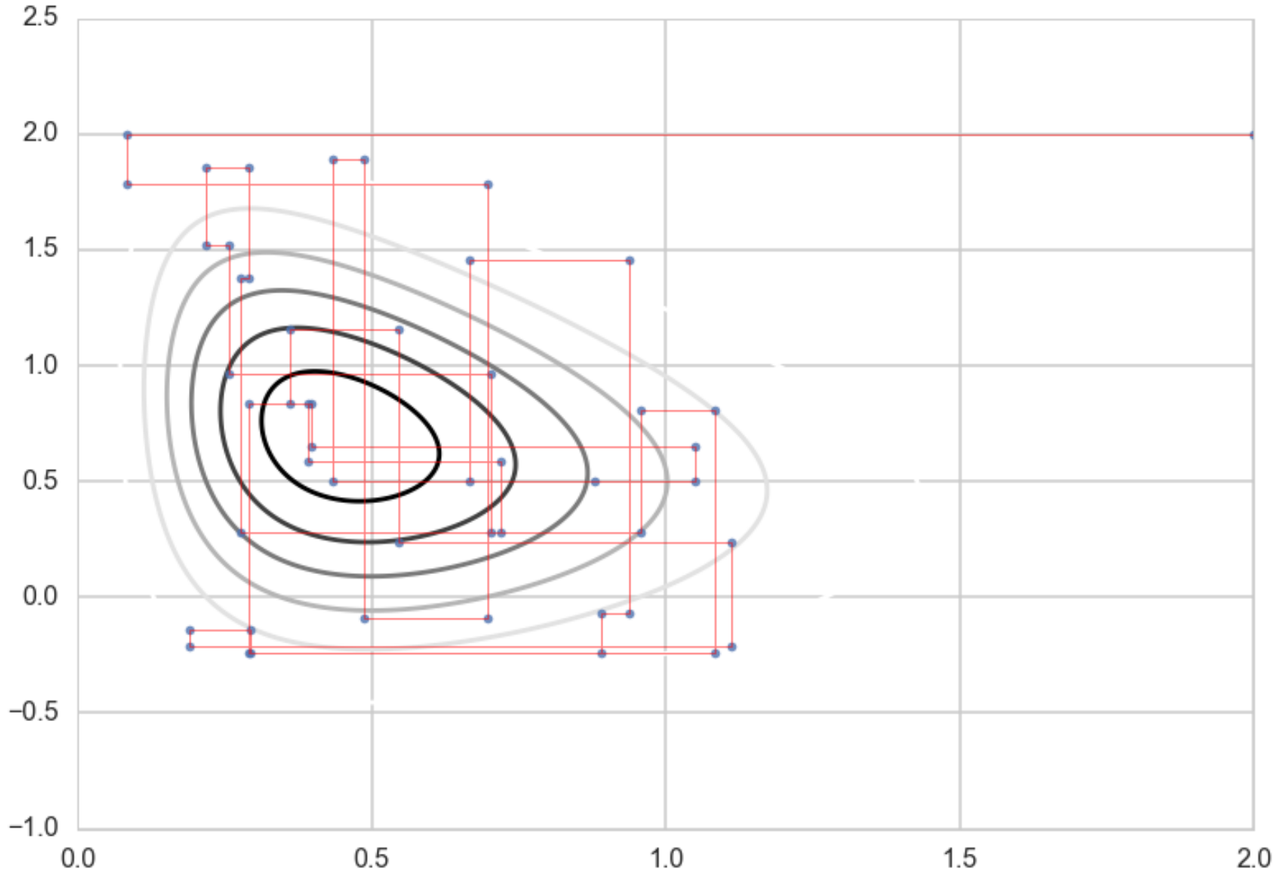


The idea of Gibbs

$$f(x_t) = \int h(x_t, x_{t-1}) f(x_{t-1}) dx_{t-1}$$, a Stationary distribution.

$h(x, x') = \int dy f(x|y) f(y|x')$.: Sample alternately to get transitions.

Can sample x marginal and $x|y$ so can sample the joint x, y.



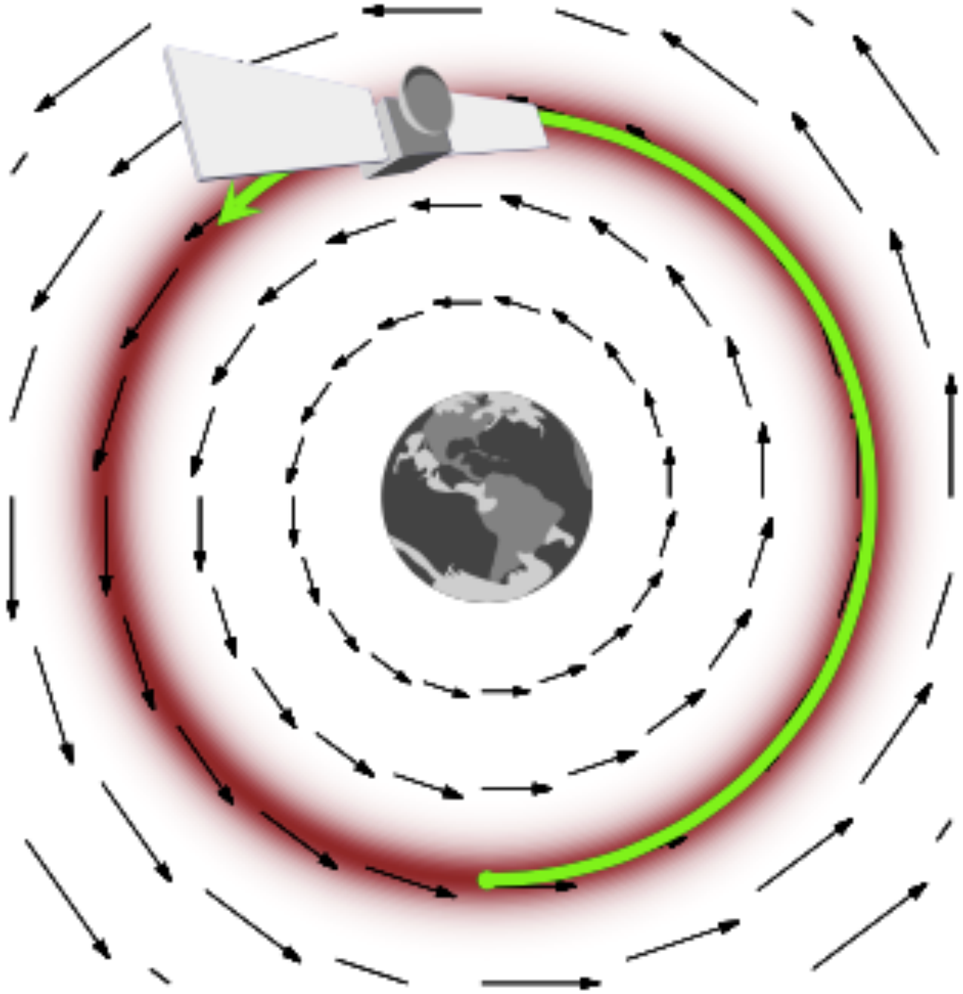
Data Augmentation

The difference from Gibbs Sampling: the other variable, say y, is to be treated as latent.

The game is to construct a joint $p(x, y)$ such that we can sample from $p(x|y)$ and $p(y|x)$, and then find the marginal

$$p(x) = \int dy\, p(x, y).$$

HMC: need glide



DATA AUGMENTATION: with an additional momentum gives energy

Hamiltonian $H(p,q) = \frac{p^2}{2m} + V(q)$

Hamiltonian flow: reversible, time-invariant, volume-preserving

Thrusters fire away

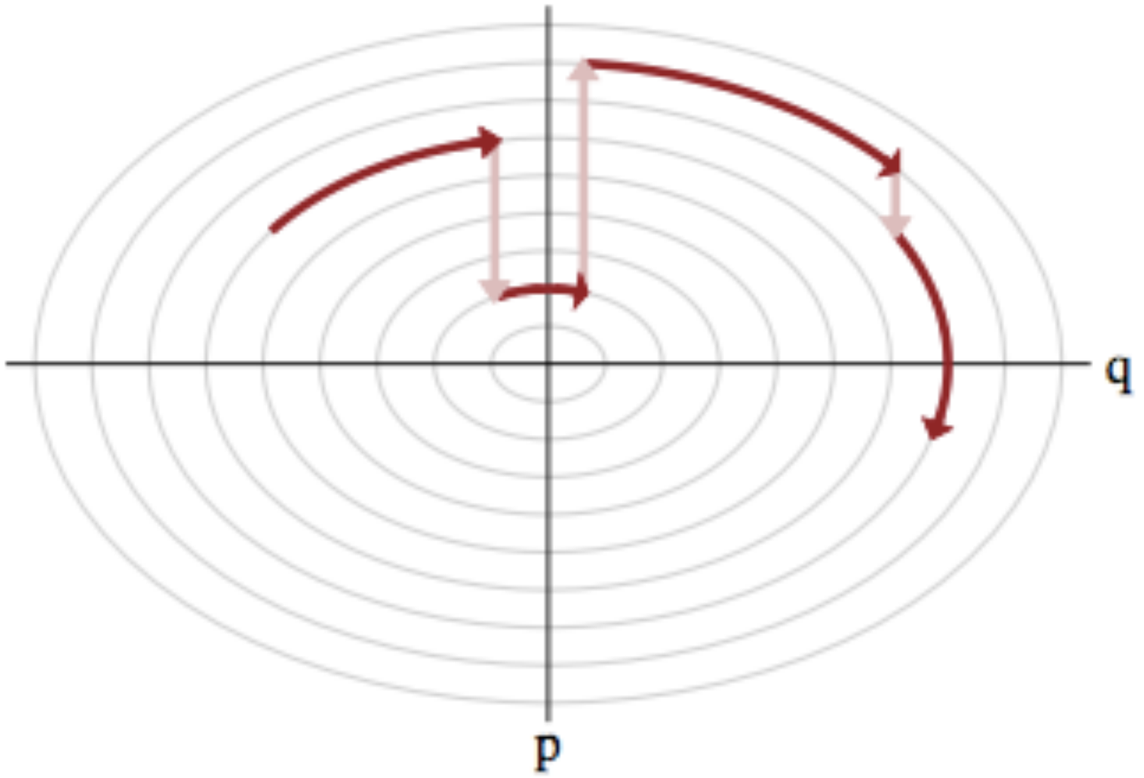
$$p(p,q) = e^{-H(p,q)} = e^{-K(p,q)}e^{-V(q)} = p(p|q)p(q)$$

$$H(p,q) = -log(p(p,q)) = -logp(p|q) - logp(q)$$

Choice of a kinetic energy term is choice of a conditional probability distribution over the "augmented" momentum such that:

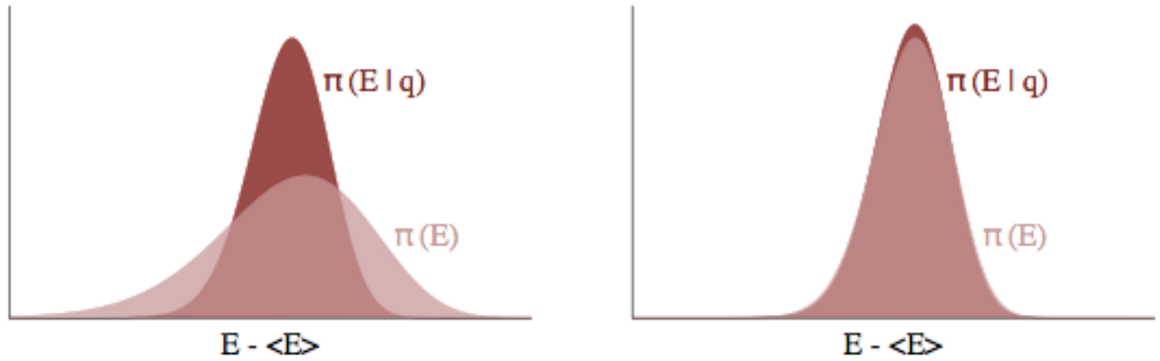
$$\int dpp(p,q) = \int dpp(p|q)p(q) = p(q)\int p(p|q)dp = p(q)$$

.



Tuning:

- The ideal kinetic energy interacts with target, in practice we often use $K(p) = p' M^{-1} p$
- Set inverse mass matrix to the covariance of the target distribution: maximally decorrelate the target. Do in warmup phase.
- use symplectic integration
- need to determine L and ϵ.
- generally static not good, under samples tails (high-energy microcanonicals). Estimate dynamically: NUTS (pymc3 and Stan)

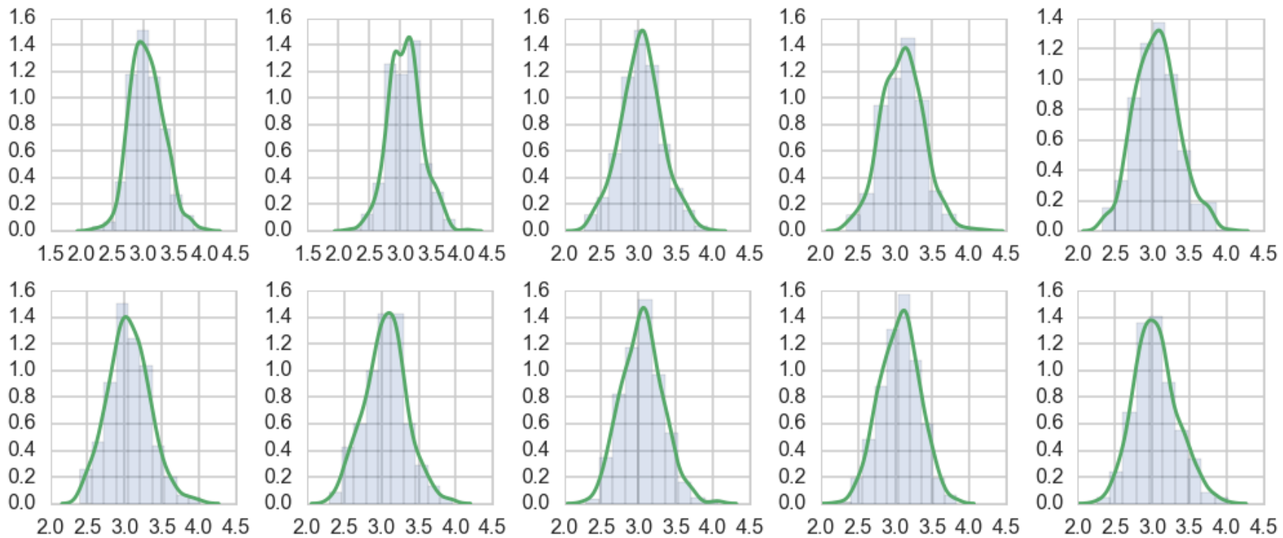


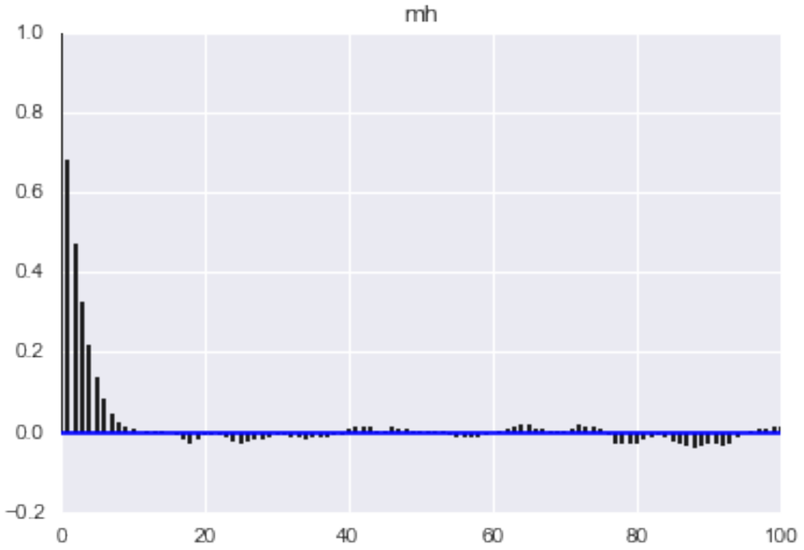
Model convergence: be paranoid

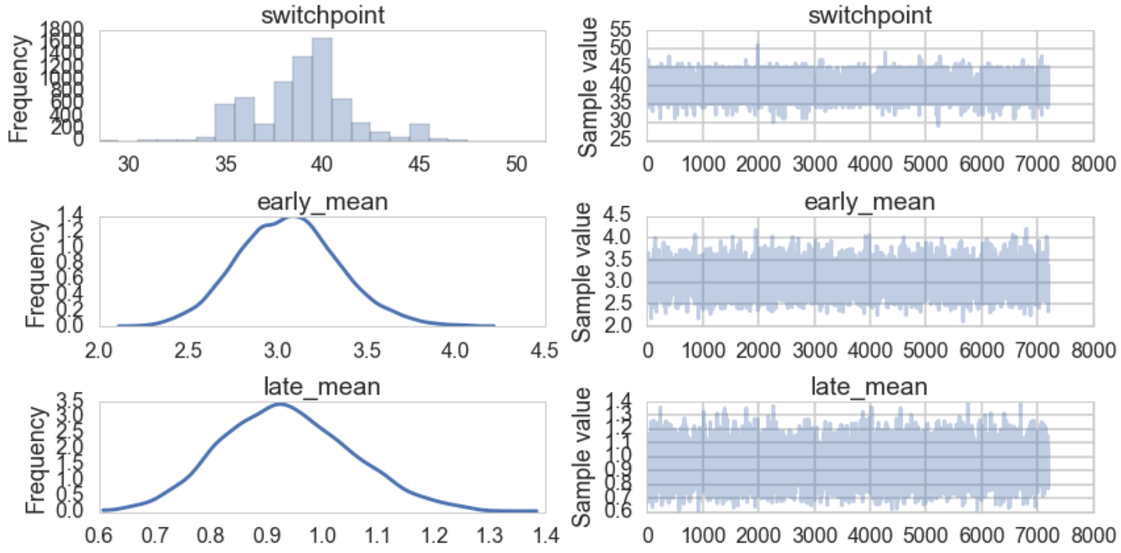
- traces white noisy
- diagnose autocorrelation, check parameter correlations

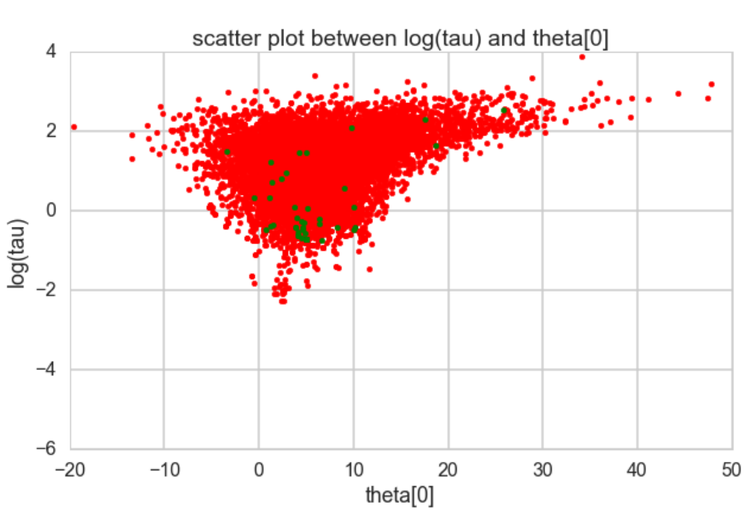
```
pm.trace_to_dataframe(trace).corr()
```

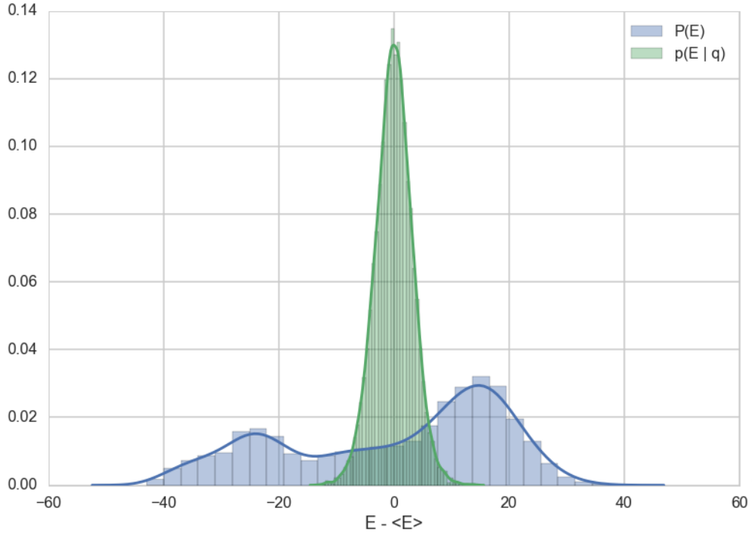
- visually inspect histogram every m samples
- traceplots from different starting points, different chains
- formal tests: Gewecke, Gelman-Rubin, Effective Sample Size, accept rate
- HMC/NUTS: check divergences, check BFMI (conditional to marginal energy ratio)












```
df=pm.trace_to_dataframe(traceni)
df.corr()
```

	sigma	mu	alpha1	alpha2
sigma	1.000000	-0.000115	-0.003153	0.003152
mu	-0.000115	1.000000	0.002844	0.008293
alpha1	-0.003153	0.002844	1.000000	-0.999938
alpha2	0.003152	0.008293	-0.999938	1.000000

STATS

WHEN BAYES

from Jim Savage

Jim Savage
@jim_savage_

Following

A test for whether a problem requires Bayesian methods:
1. Is there information that is not in your data about population-level unknowns?
2. Do you need coherent uncertainty?
3. Are you combining complex models and want uncertainty to percolate through?

Yes to any? Bayes it.

11:49 AM - 9 Apr 2018

11 Retweets **90** Likes

3 11 90

Tweet your reply

Jake Mortenson @jm0rt · 20h
Replying to @jim_savage_
Have been looking for an excuse to do Bayesian stuff in a tax policy research setting. But isn't there also 4, do you have some sparsely populated (and interesting) bins? The answer to 1 and 2 are virtually always yes, but have avoided so far because our data are typically yuge.

1

Jim Savage @jim_savage_ · 20h
I couldn't add 4) You want to generalize to new populations (post-strat) & so want to estimate sub-group effects, but your sample has small N in those sub-groups, there's a lot of value in hierarchical priors.

Are we saying the same thing?

Jake Mortenson @jm0rt · 19h
That was part of my point, the other part being (perhaps out of my depth): with large data the benefits from incorporating priors may not be large (fixed effects may be sufficient, depending on parameters of interest), and also computation might be time-expensive. Sound right?

1 1

Jim Savage @jim_savage_ · 19h
See rule 1 though: if there is information your enormous data doesn't contain about the unknown of interest (in the population--which for most purposes is a future population) then there might still be value in having priors. Turkey before thanksgiving story.

1

Noah Motion @statmodcitizen · 22h
Replying to @jim_savage_
My intuition is that the answer to (2) is always "yes", but I may be misunderstanding what you mean by the question...

1

Jim Savage @jim_savage_ · 22h
Strictly yes, if computation and analyst time has no cost. Business maximize profit, not correctness.

4

Frank Harrell @f2harrell · 18h
Replying to @jim_savage_
Nice. I'd simply say "Does your problem require statistical inference?". If yes, Bayes it. Among other things this solves is that inference is exact. Most frequentist analyses are approximations, other than the ordinary linear model and a few others.

11

Latent Variables

- dont think of bayes/frequentist, think of observed x /Latent z
- anything unobserved is latent (this is the posterior predictive point of view, x as θ), thus standard bayesian viewpoint: nuisance parameters are latent
- latent factors in matrix factorization, mixtures, recommendations...cluster zs

Generative model

$$p(x, z) = p(x|z)p(z)$$

- **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**.

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: $posterior = \frac{likelihood \times prior}{evidence}$

- prior-predictive = evidence: $p(y) = E_{p(\theta)}[\mathcal{L}] = \int d\theta p(y|\theta)p(\theta)$ a normalization, useful for workflow and EB
- What if θ is multidimensional? Marginal posterior: $p(\theta_1|D) = \int d\theta_{-1} p(\theta|D).$
- posterior predictive: the distribution of a future data point y^*:

$$p(y^*|D = \{y\}) = E_{p(\theta|D)}[p(y|\theta)] = \int d\theta p(y^*|\theta)p(\theta|\{y\}).$$

Marginalization

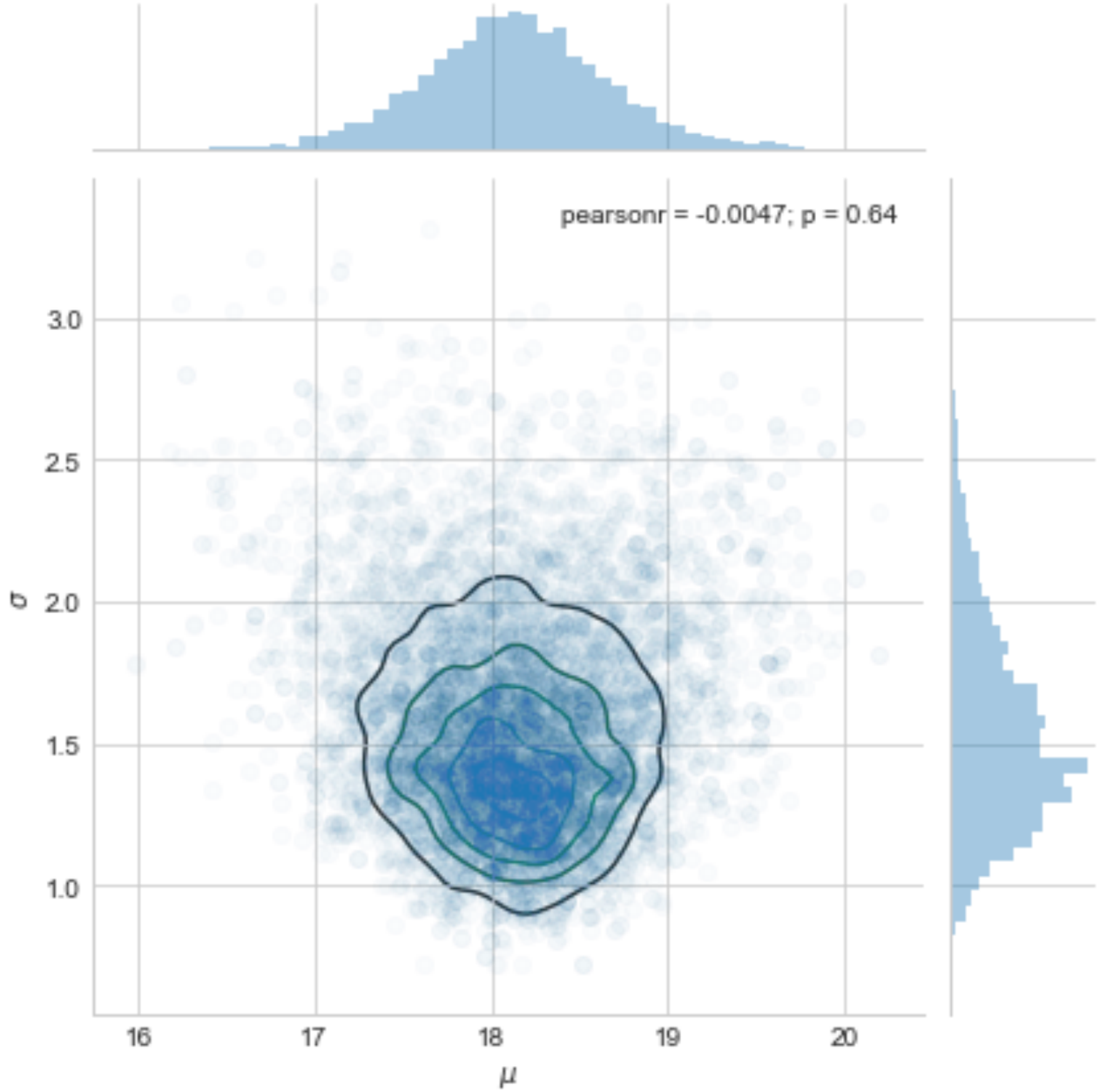
Marginal posterior: $p(\theta_1|D) = \int d\theta_{-1} p(\theta|D)$.

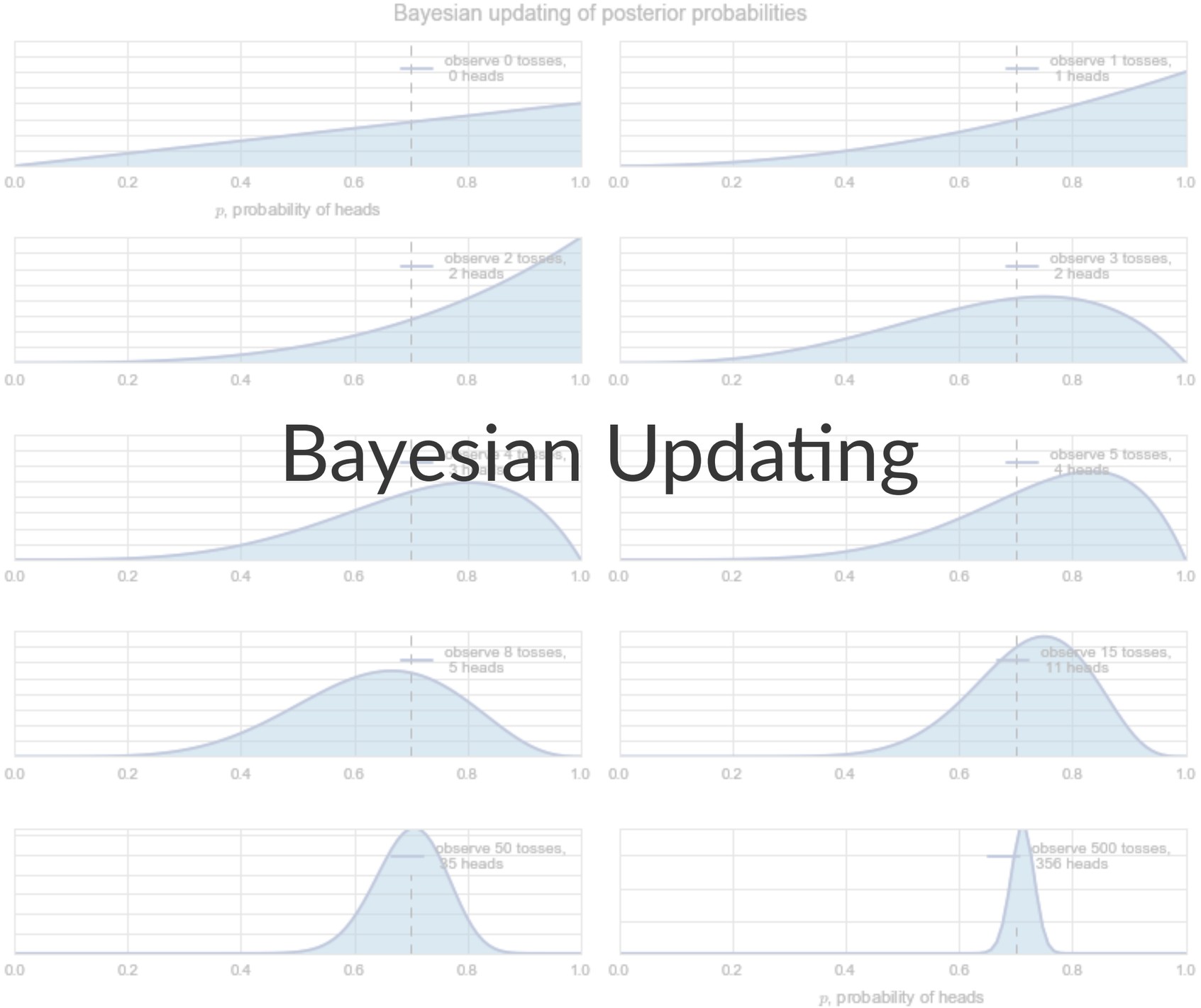
```
samps[20000::,:].shape #(10001, 2)
```

```
sns.jointplot(
    pd.Series(samps[20000::,0], name="$\mu$"),
    pd.Series(samps[20000::,1], name="$\sigma$"),
    alpha=0.02)
    .plot_joint(
        sns.kdeplot,
    zorder=0, n_levels=6, alpha=1)
```

Marginals are just 1D histograms

```
plt.hist(samps[20000::,0])
```





Bayesian Updating

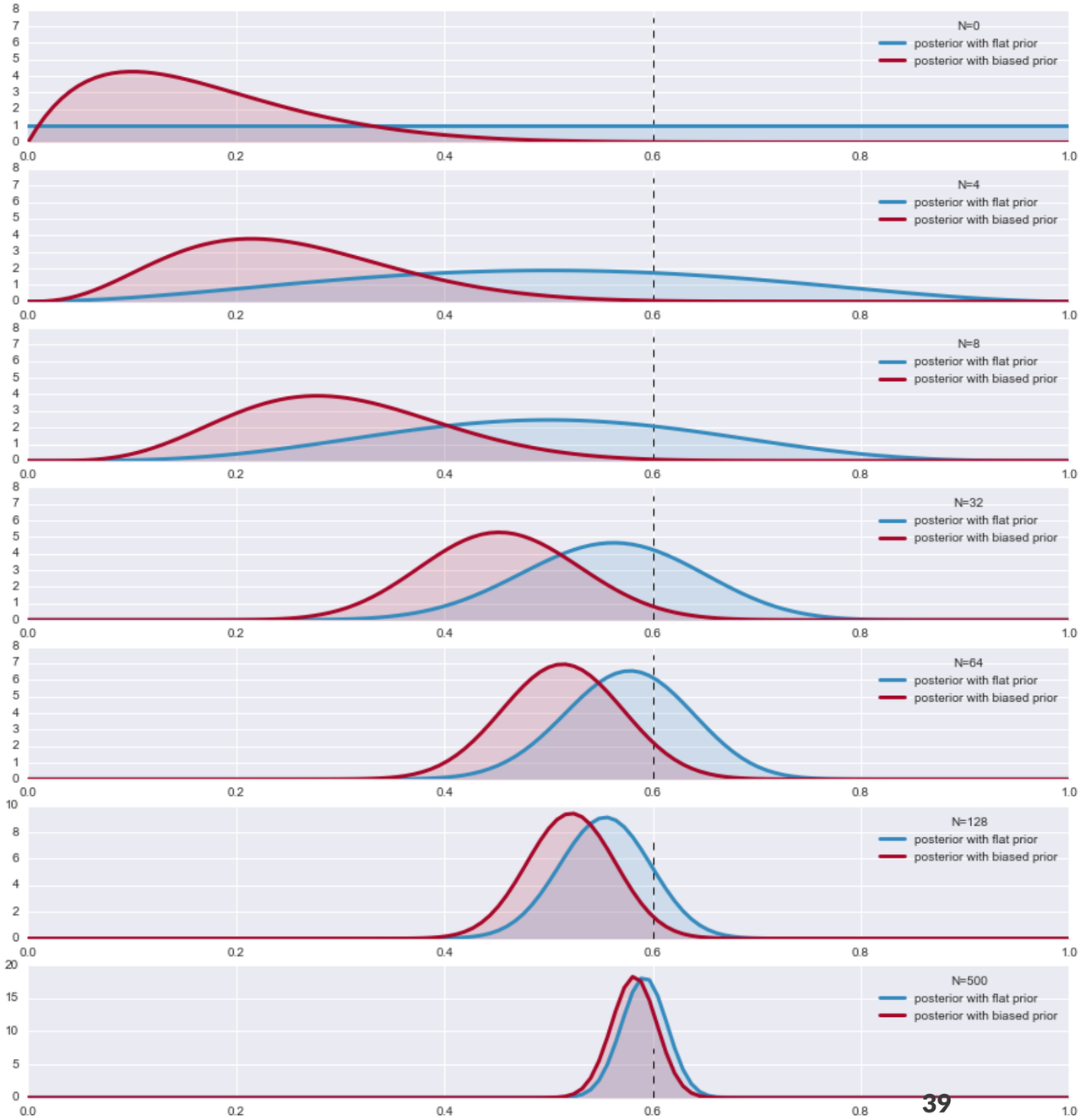
Data Overwhelms priors

Define $\kappa = \sigma^2/\tau^2$

$$\mu_p = \frac{b}{a} = \frac{\kappa}{\kappa+n}\hat{\mu} + \frac{n}{\kappa+n}\bar{y}$$

$$\frac{1}{\tau_p^2} = \frac{1}{\tau^2} + \frac{n}{\sigma^2}.$$

- priors regularize data for small data
- but large data overwhelms priors



Exchangeability

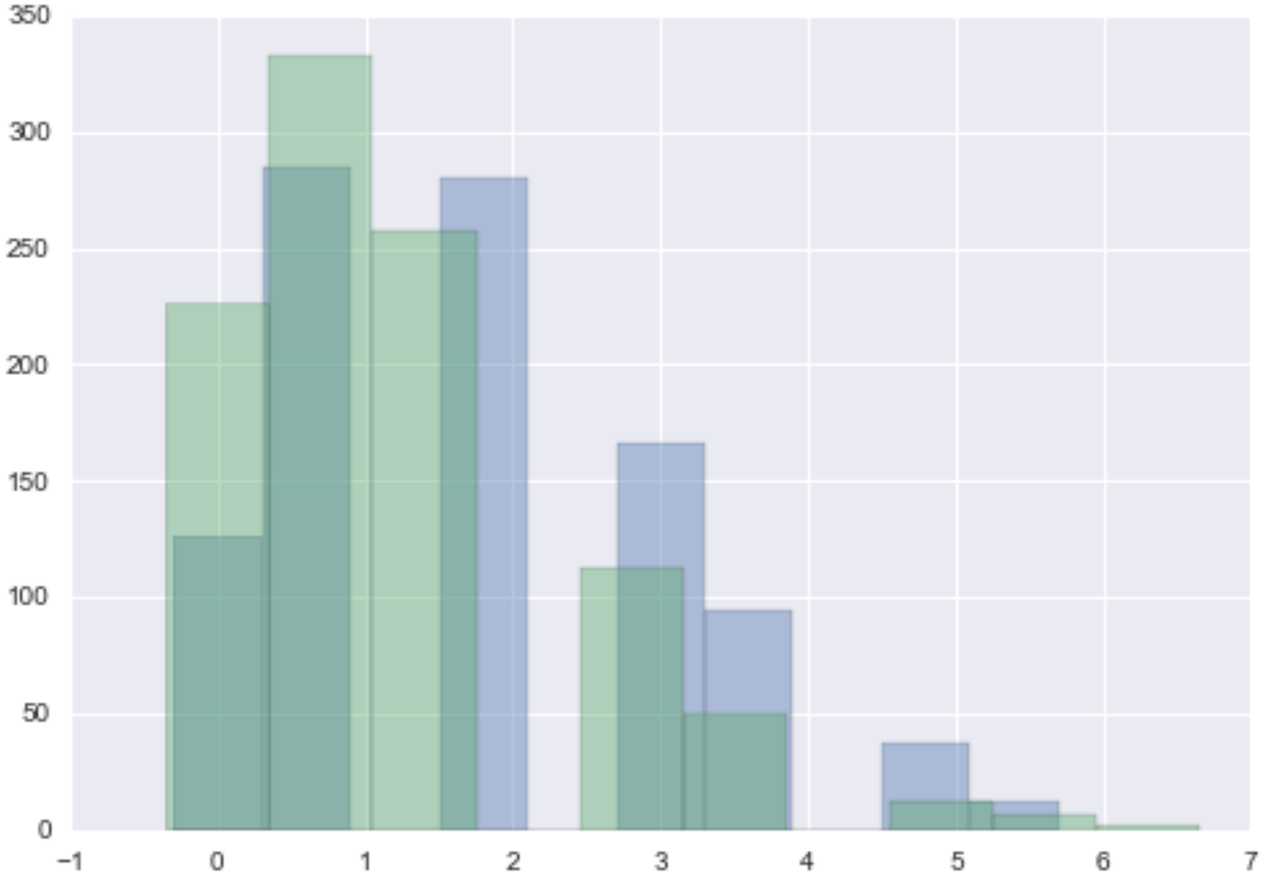
Lets assume that the number of children of a women in any one of these classes can me modelled as coming from ONE birth rate.

The in-class likelihood for these women is invariant to a permutation of variables.

This is really a statement about what is IID and what is not.

It depends on how much knowledge you have...

Posterior Predictives



$$p(y^*|D) = \int d\theta p(y^*|\theta)p(\theta|D)$$

Sampling easy (mothers poisson-gamma):

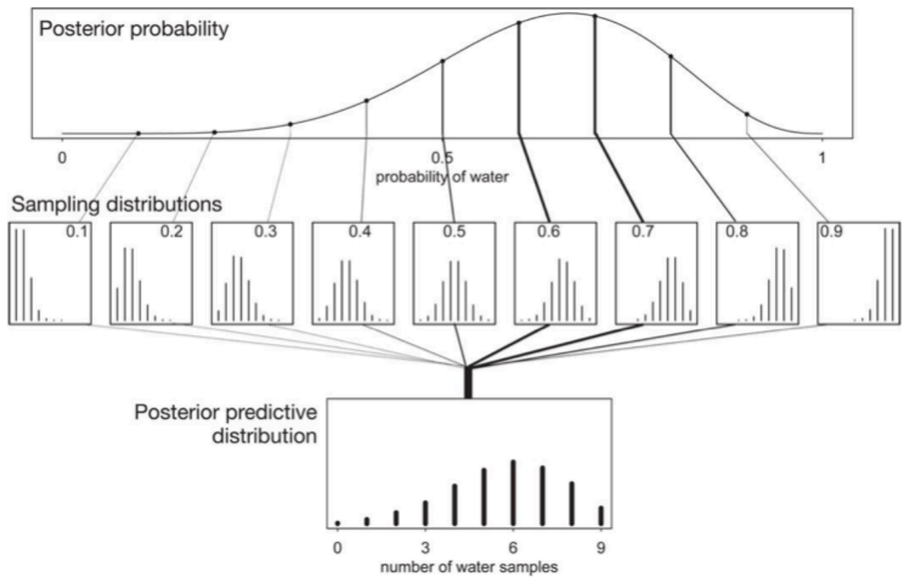
```
postpred1 = poisson.rvs(theta1trace)
postpred2 = poisson.rvs(theta2trace)
```

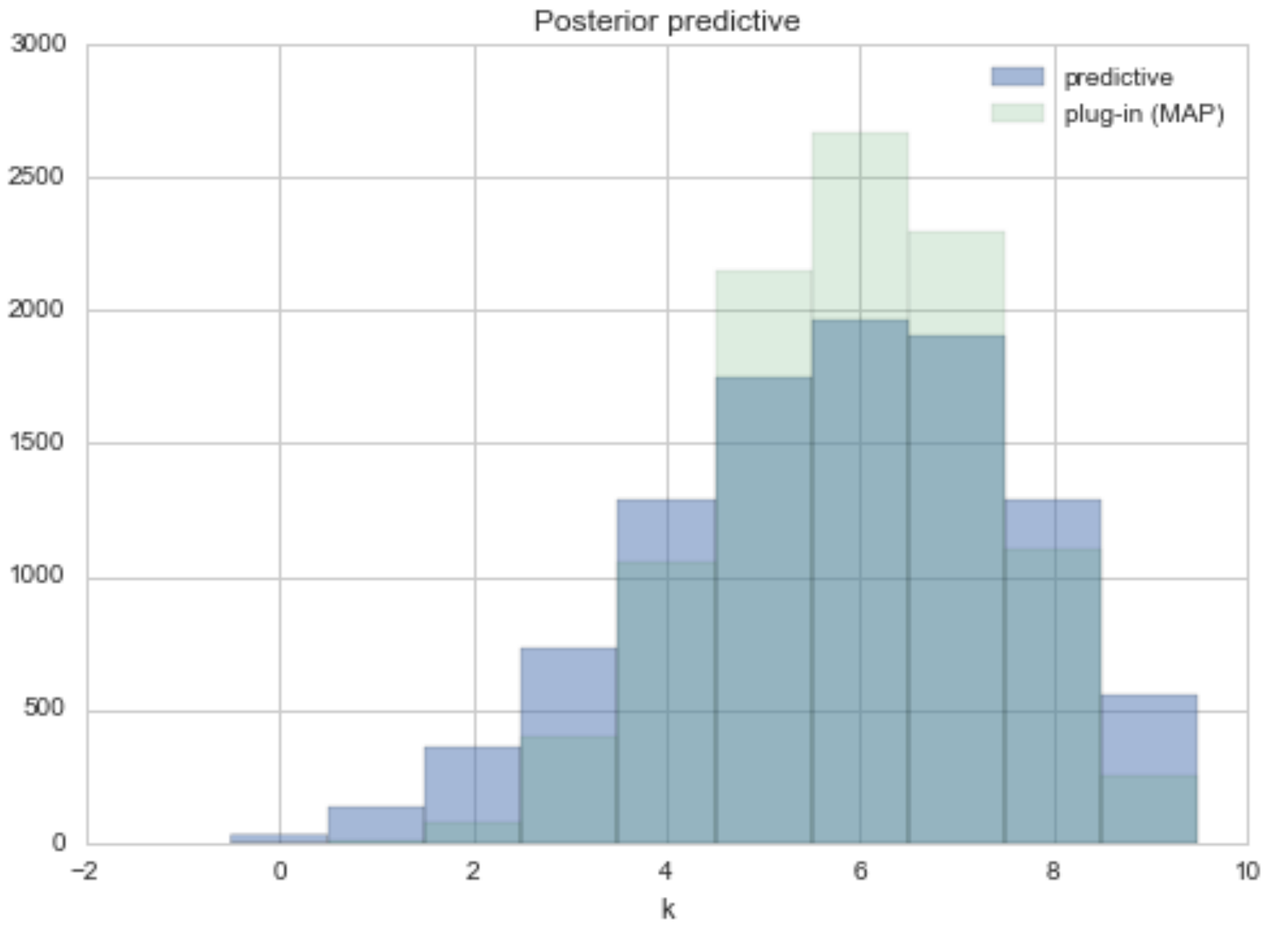
Exact: Negative Binomial (requires math):

$$E[y^*] = \frac{(a + \sum y_i)}{(b+N)}$$

$$var[y^*] = \frac{(a + \sum y_i)}{(b+N)^2}(N + b + 1).$$

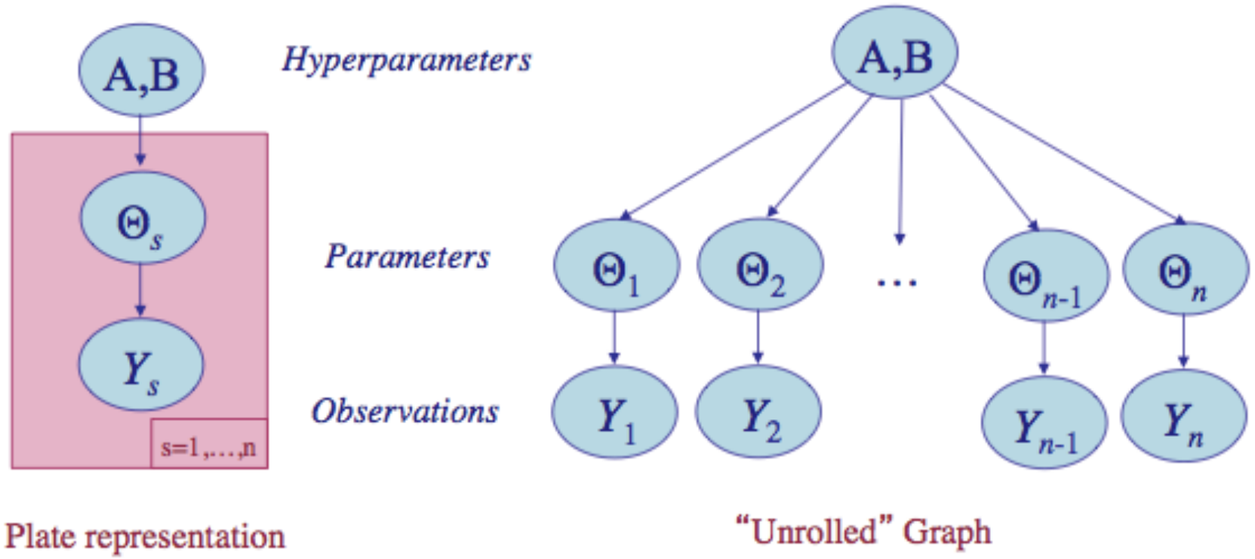
Posterior Predictive Smear





pp vs sampling distrib at MAP →

Partial pooling: Hierarchical Model



θ_is drawn from "population distribution" given by a conjugate Beta prior $Beta(\alpha, \beta)$ with **hyperparameters** α and β.

$$\theta_i \sim Beta(\alpha, \beta).$$

$$p(\Theta|\alpha, \beta) = \prod_{i=1}^{70} Beta(\theta_i, \alpha, \beta).$$

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.

Empirical Bayes or Type-2 Likelihood

Posterior-predictive distribution, as a function of upper level parameters $\eta = (\alpha, \beta)$.

$$p(y^*|D, \eta) = \int d\theta \, p(y^*|\theta) \, p(\theta|D, \eta)$$

A likelihood with parameters η and simply use maximum-likelihood with respect to η to estimate these η using our "data" y^*

Used in GPs, even can be sampled from

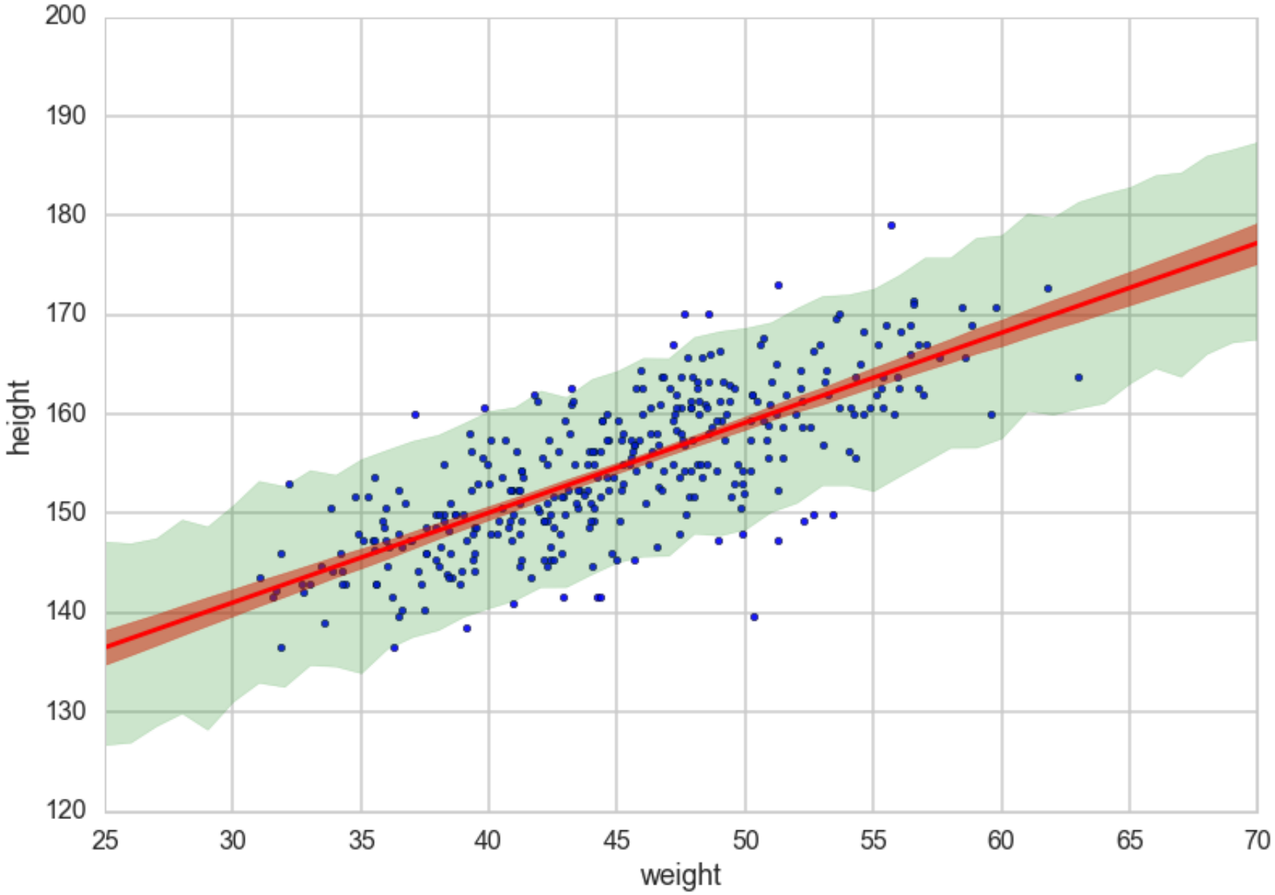
Levels of Bayes

Method	Definition
Maximum Likelihood	$\hat{\theta} = argmax_\theta p(D\|\theta)$
MAP estimation	$\hat{\theta} = argmax_\theta p(D\|\theta)p(\theta\|\eta)$
ML-2 (Empirical Bayes)	$\hat{\eta} = argmax_\eta \int d\theta\, p(D\|\theta)p(\theta\|\eta) = argmax_\eta p(D\|\eta)$
MAP-2	$\hat{\eta} = argmax_\eta \int d\theta\, p(D\|\theta)p(\theta\|\eta)p(\eta) = argmax_\eta p(D\|\eta)p(\eta)$
Full Bayes	$p(\theta, \eta\|D) \propto p(D\|\theta)p(\theta\|\eta)p(\eta)$

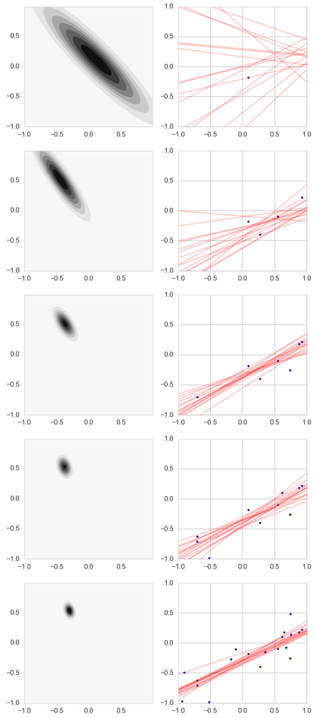
Howto Sampling

- a DAG, with observations at the bottom of a tree, next layer intermediate parameters, upper layers hyper-parameters
- sample conditionals from parents up the tree.
- general structure is sampling steps inside Gibbs
- stan, pymc3 all have this structure

Bayesian Regression



- posterior narrower (μ spread) than PP
- supervised learning, a distrib at each x



Model reparametrization helps samplers

- make parameters identifiable
- center covariates
- in hierarchical models, try and compress the hierarchy. Eg gelman schools reparametrization trick

glms

- linear regression with a link. likelihoods chosen MAXENT

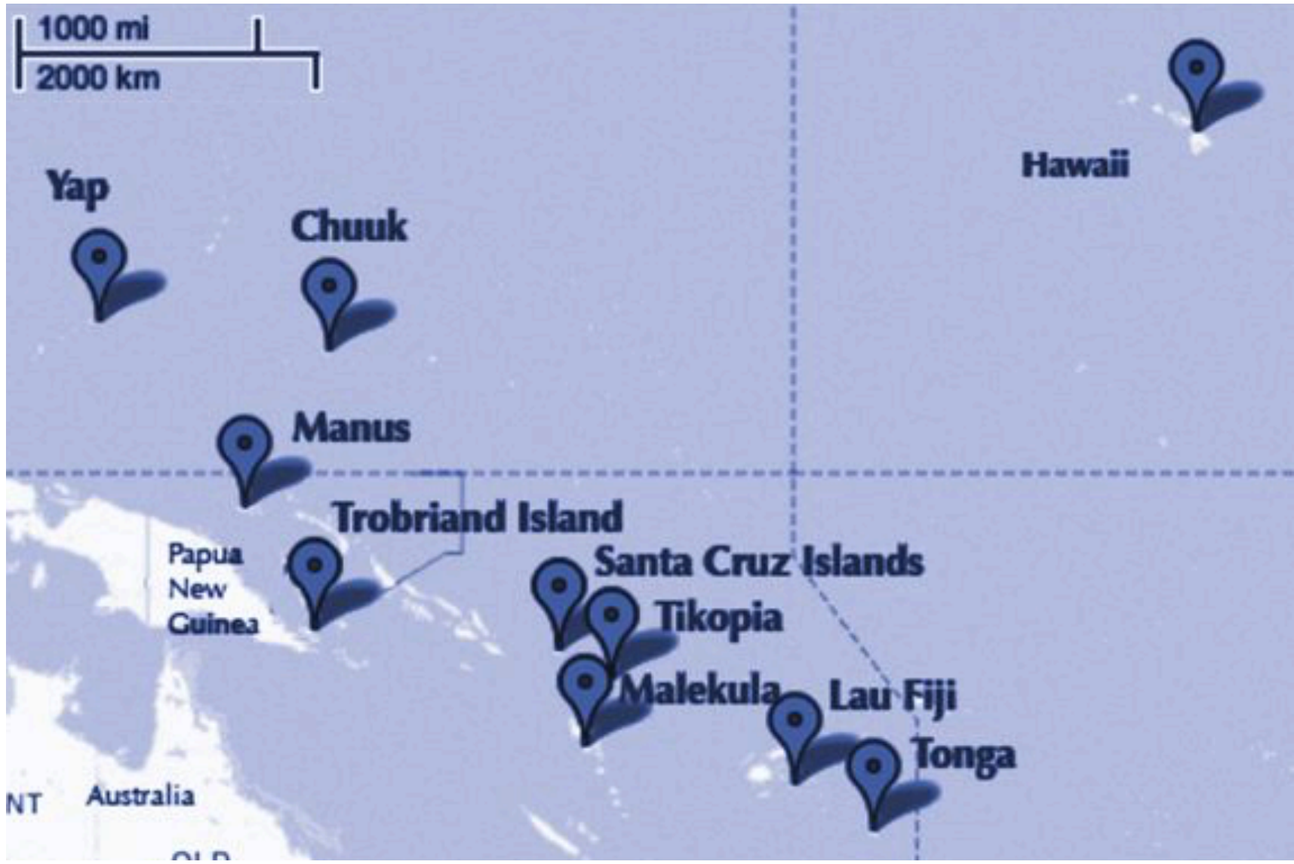
  $f(p_i) = \alpha + \beta x_i$ where p_i is the parameter at the ith data point.

For most GLMs, the common links we use are the *logit* link to model the space of probabilities, and the *log* link which you will use here to enforce positiveness on a parameter.

Oceanic tools

From Mcelreath:

The island societies of Oceania provide a natural experiment in technological evolution. Different historical island populations possessed tool kits of different size. These kits include fish hooks, axes, boats, hand plows, and many other types of tools. A number of theories predict that larger populations will both develop and sustain more complex tool kits. So the natural variation in population size induced by natural variation in island size in Oceania provides a natural experiment to test these ideas. It's also suggested that contact rates among populations effectively increase population size, as it's relevant to technological evolution. So variation in contact rates among Oceanic societies is also relevant. (McElreath 313)



Model M1

	culture	population	contact	total_tools	mean_TU	logpop	clevel
0	Malekula	1100	low	13	3.2	7.003065	0
1	Tikopia	1500	low	22	4.7	7.313220	0
2	Santa Cruz	3600	low	24	4.0	8.188689	0
3	Yap	4791	high	43	5.0	8.474494	1
4	Lau Fiji	7400	high	33	5.0	8.909235	1
5	Trobriand	8000	high	19	4.0	8.987197	1
6	Chuuk	9200	high	40	3.8	9.126959	1
7	Manus	13000	low	28	6.6	9.472705	0
8	Tonga	17500	high	55	5.4	9.769956	1
9	Hawaii	275000	low	71	6.6	12.524526	0

$$
\begin{aligned}
T_i &\sim Poisson(\lambda_i) \\
log(\lambda_i) &= \alpha + \beta_P log(P_i) + \beta_C C_i + \beta_{PC} C_i log(P_i) \\
\alpha &\sim N(0, 100) \\
\beta_P &\sim N(0, 1) \\
\beta_C &\sim N(0, 1) \\
\beta_{PC} &\sim N(0, 1)
\end{aligned}
$$

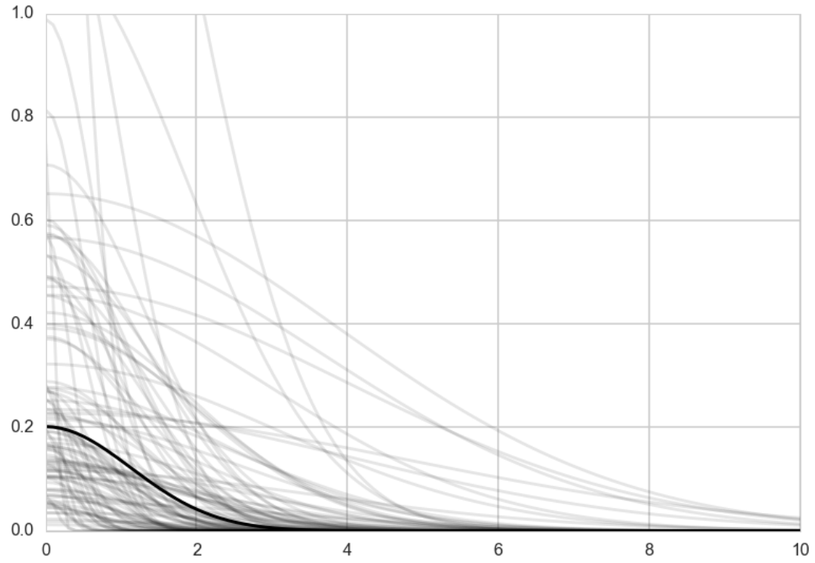
```
with pm.Model() as m1:
    betap = pm.Normal("betap", 0, 1)
    betac = pm.Normal("betac", 0, 1)
    betapc = pm.Normal("betapc", 0, 1)
    alpha = pm.Normal("alpha", 0, 100)
    loglam = alpha + betap*df.logpop +
        betac*df.clevel + betapc*df.clevel*df.logpop
    y = pm.Poisson("ntools", mu=t.exp(loglam), observed=df.total_tools)
```

```
with m1:
    trace=pm.sample(10000, njobs=2)
Average ELBO = -55.784:
100%|██████████| 200000/200000 [00:15<00:00, 13019.16it/s]   12683.03it/s]
100%|██████████| 10000/10000 [01:59<00:00, 83.80it/s]
```

Oceanic Tools Correlations: Example of GP

We modeled society specific intercepts for oceanic tools as draws from a 0 mean multivariate gaussian and correlation function depending on distance: nearer societies have similar intercepts.

Covariance posteriors:



$$T_i \sim \text{Poisson}(\lambda_i)$$

$$\log \lambda_i = \alpha + \gamma_{\text{SOCIETY}[i]} + \beta_P \log P_i$$

$$\gamma \sim \text{MVNormal}\big((0, \ldots, 0), \mathbf{K}\big)$$

$$\mathbf{K}_{ij} = \eta^2 \exp(-\rho^2 D_{ij}^2) + \delta_{ij}(0.01)$$

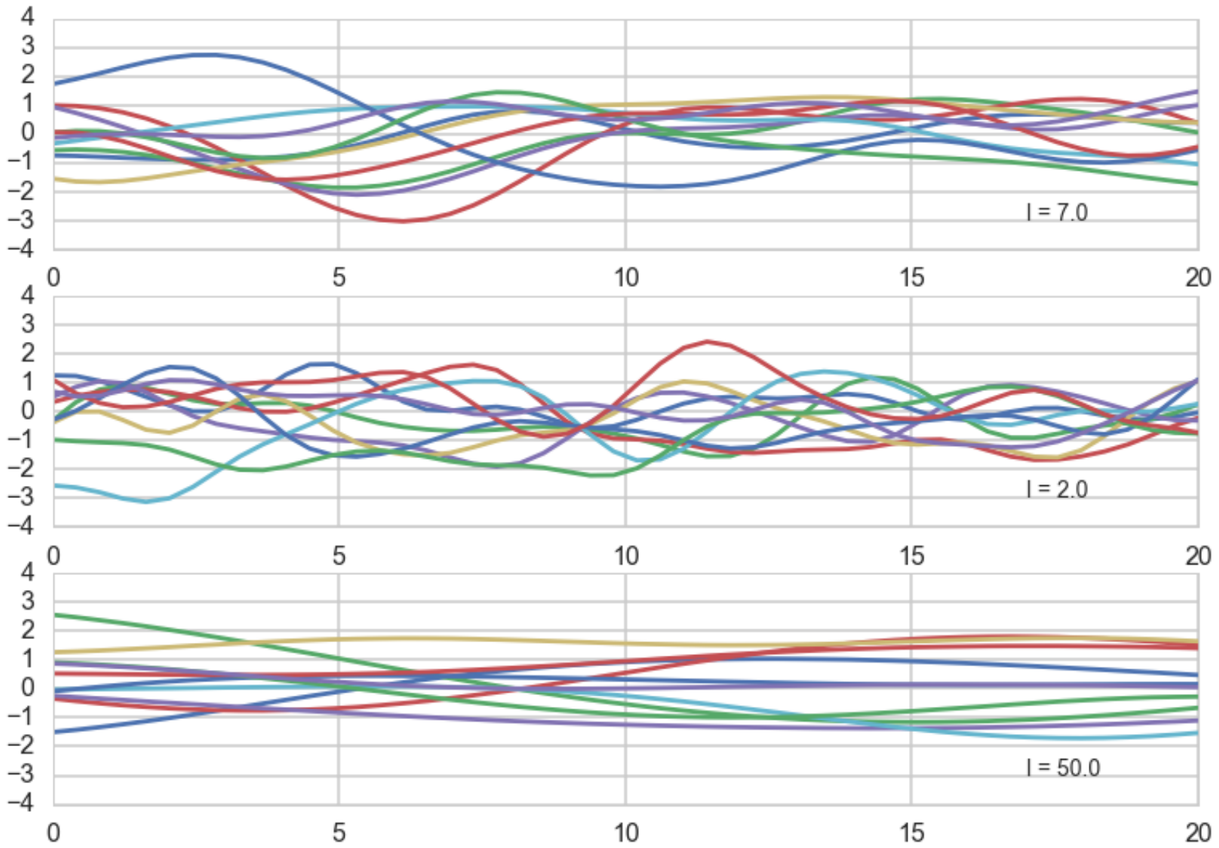
$$\alpha \sim \text{Normal}(0, 10)$$

$$\beta_P \sim \text{Normal}(0, 1)$$

$$\eta^2 \sim \text{HalfCauchy}(0, 1)$$

$$\rho^2 \sim \text{HalfCauchy}(0, 1)$$

Generating curves from a kernel-based covariance



a Gaussian Process defines a prior distribution over functions!

Once we have seen some data, this prior can be converted to a posterior over functions, thus restricting the set of functions that we can use based on the data.

KEY INSIGHT:

MARGINAL IS DECOUPLED

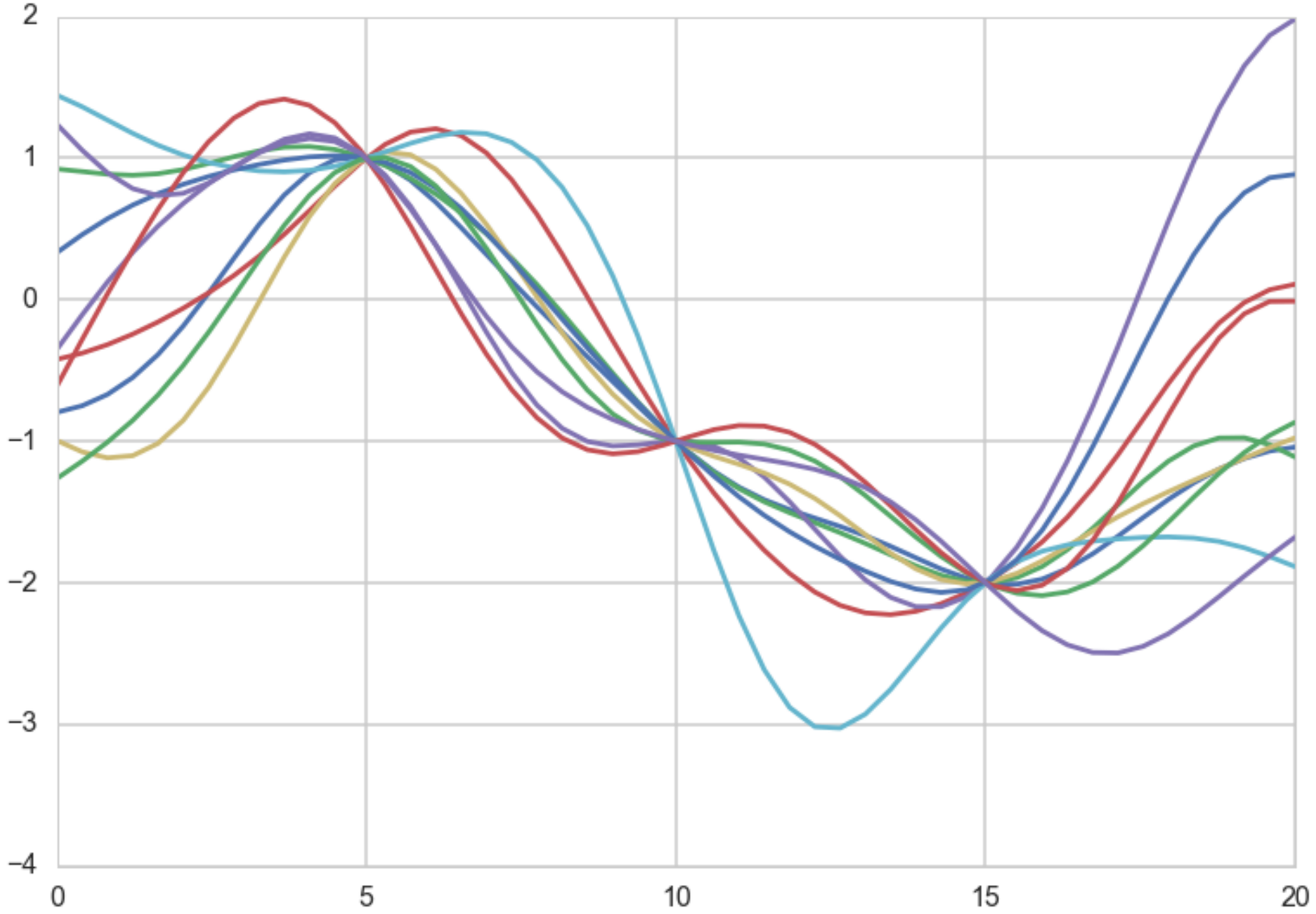
...for the marginal of a gaussian, only the covariance of the block of the matrix involving the unmarginalized dimensions matters! Thus "if you ask only for the properties of the function (you are fitting to the data) at a finite number of points, then inference in the Gaussian process will give you the same answer if you ignore the infinitely many other points, as if you would have taken them all into account!"

-Rasmunnsen

Conditional

$$p(f^* \mid y) = \mathcal{N}\left(\mu_* + K_*(K + \sigma^2 I)^{-1}(y - \mu),\ K_{**} - K_*(K + \sigma^2 I)^{-1} K_*^T\right)$$

EQUALS Predictive


2
1
0
−1
−2
−3
−4
0
5
10
15
20

MODEL CHECKING

Multiple replications of the posterior predictive

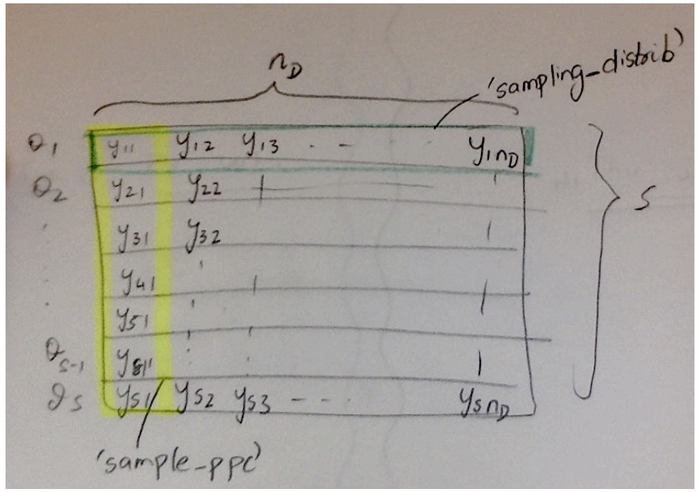
$$p(\{y^*\}) = \int p(\{y^*\}|\theta)p(\theta|\mathcal{D})d\theta, \text{ observed data: } \mathcal{D} = \{y\}$$

Replicated Data: $\{y_r\}$: data seen tomorrow if experiment replicated with same model and value of θ producing todays data $\{y\}$.

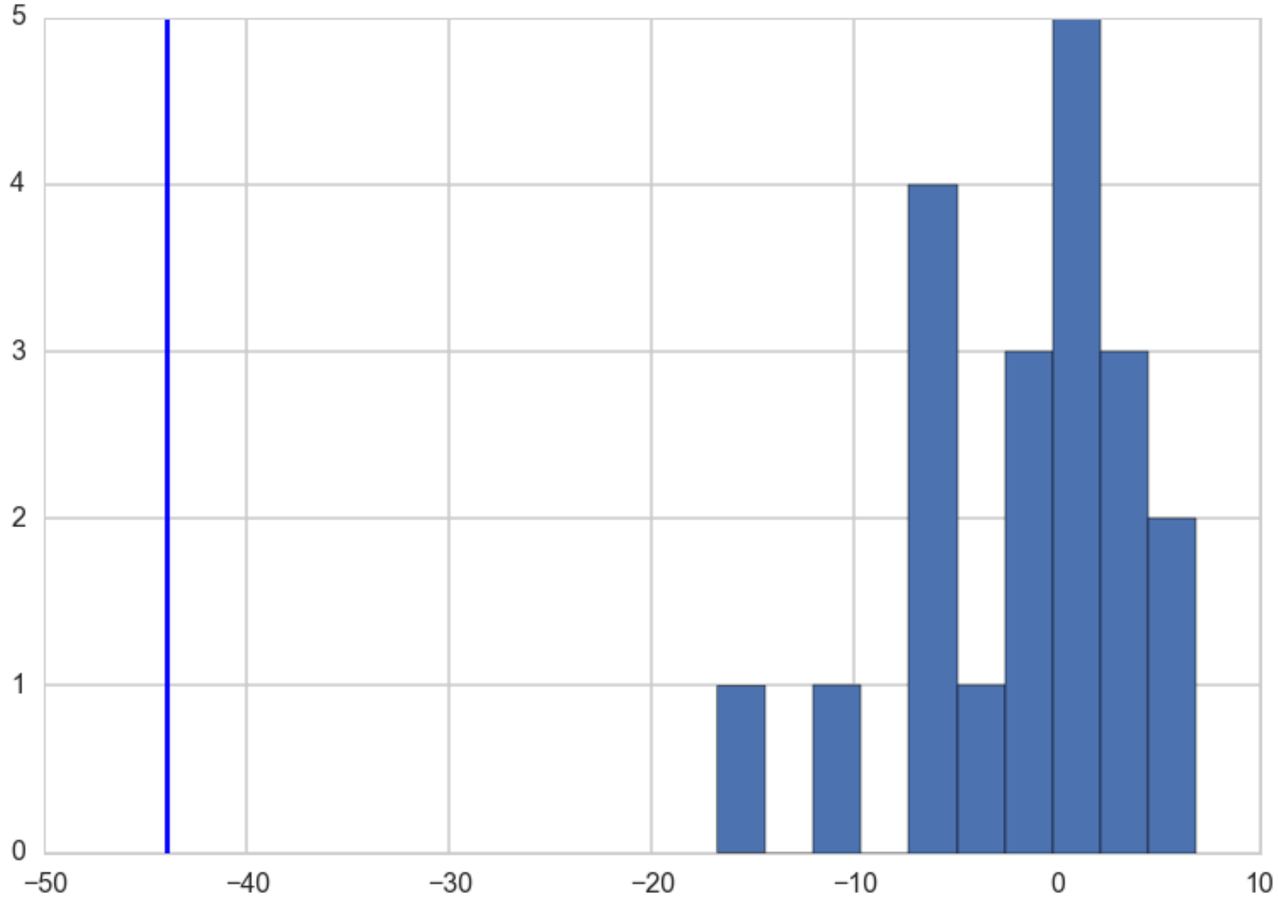
$\{y_r\}$ comes from posterior predictive, and if there are covariates $\{x^*\}$, then $\{y_r\}$ is calculated at those covariates only (`sample_ppc`).

Departure from usual predictive sampling

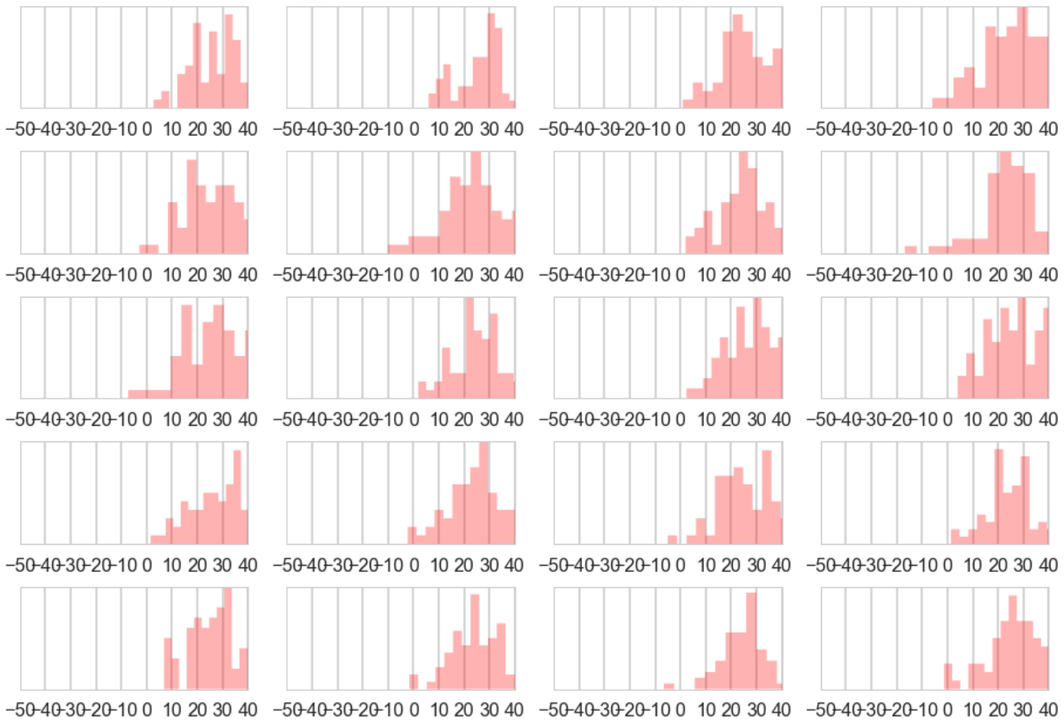
Sample an entire $\{y_r\}$ at each θ from trace.

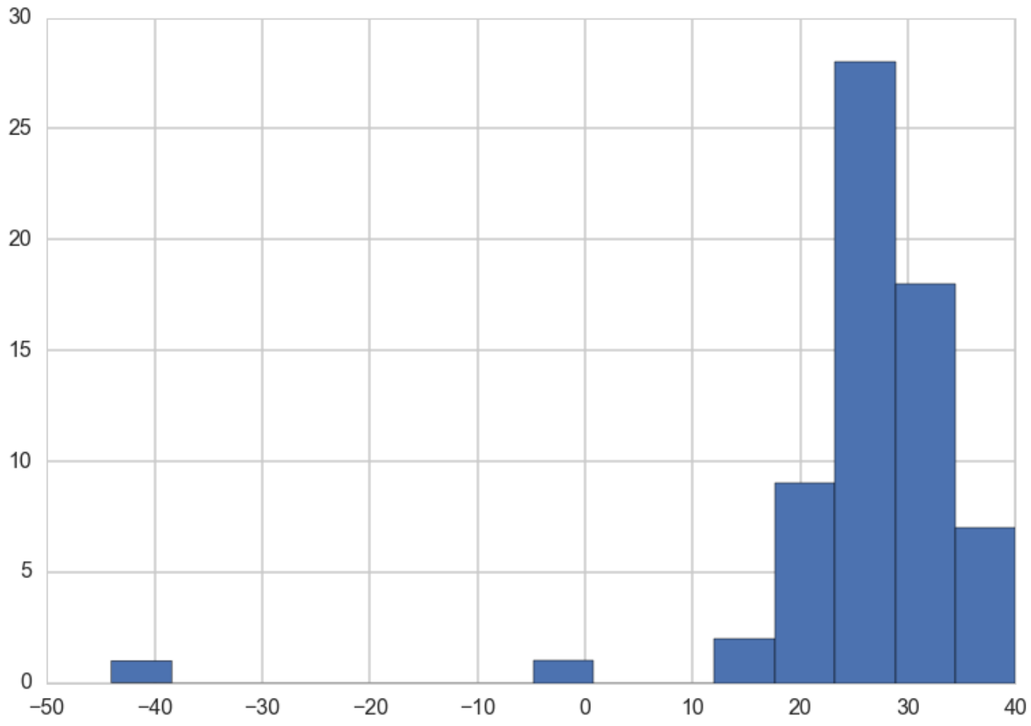


For example the minimum value of speed of light in 20 predictive replications.



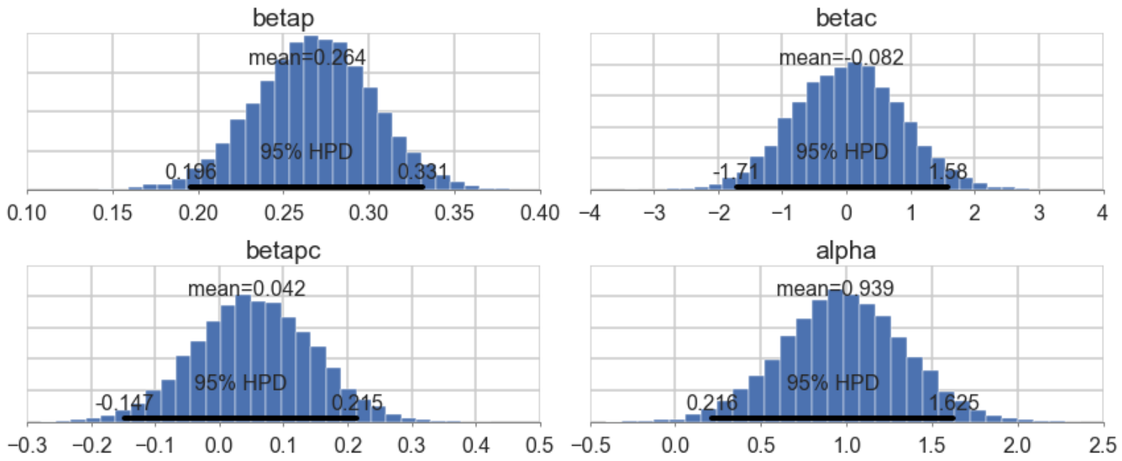
Visual Checking

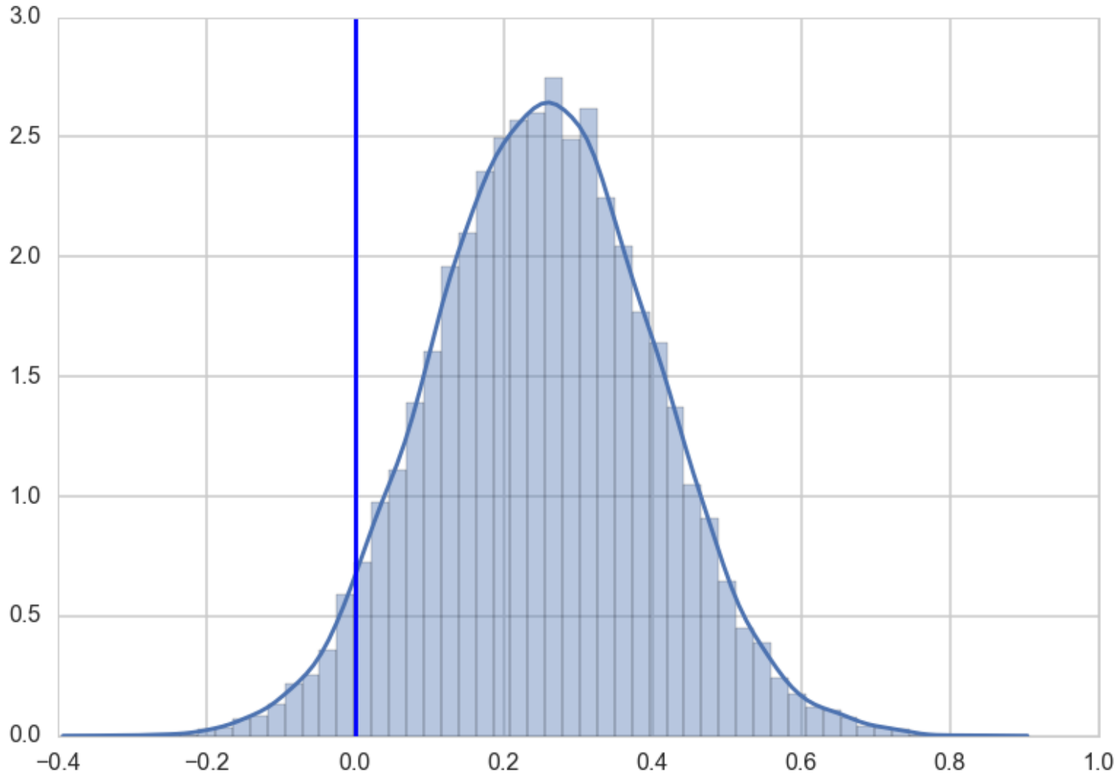




Do these even look similar??

Oceanic Tools counterfactual checking





MODEL COMPARISON AND ENSEMBLING

Model comparison

The key idea in model comparison is that we will sort our average utilities in some order. The exact values are not important, and may be computed with respect to some true distribution or true-belief distribution M_{tb}.

$$\bar{u}(M_k, \hat{a}_k) = \int dy^* u(\hat{a}_k, y^*) p(y^*|D, M_{tb})$$

where $\hat{a}_k$ is the optimal prediction under the model M_k. Now we compare the actions, that is, we want:

$$\hat{M} = \arg\max_k \bar{u}(M_k, \hat{a}_k)$$

No calibration, but calculating the standard error of the difference can be used to see if the difference is significant, as we did with the WAIC score

Deviance

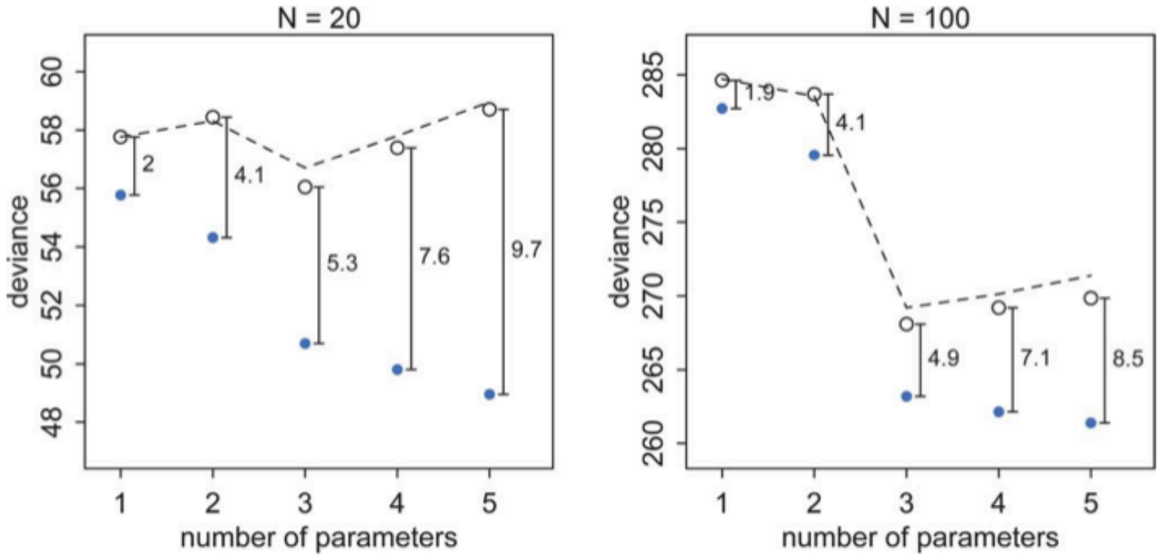
$$D(q) = -2 \sum_i log(q_i),$$

then

$$D_{KL}(p, q) - D_{KL}(p, r) = \frac{2}{N}(D(q) - D(r))$$

More generally: $D(q) = -\frac{N}{2} E_p[log(q)]$

AIC



Akaike Information Criterion, or AIC:

$$AIC = D_{train} + 2p$$

$$D_{train} = -2 * log(p(y|\theta_{mle})$$

- multivariate gaussian posterior
- flat priors
- data >> parameters

Bayesian deviance

- $D(q) = -\frac{N}{2} E_p[log(pp(y))]$ posterior predictive for points y on the test set or future data
- replace joint posterior predictive over new points y by product of marginals: ELPD: $\sum_i E_p[log(pp(y_i))]$
- Since we do not know the true distribution p, replace elpd: $\sum_i E_p[log(pp(y_i))]$ by the computed "log pointwise predictive density" (lppd) **in-sample**

$$\sum_j log \langle p(y_j|\theta) \rangle = \sum_j log \left(\frac{1}{S} \sum_s p(y_j|\theta_s) \right)$$

WAIC

$$WAIC = lppd + 2p_W$$

where

$$p_W = 2\sum_i (log(E_{post}[p(y_i|\theta)] - E_{post}[log(p(y_i|\theta))])$$

Once again this can be estimated by

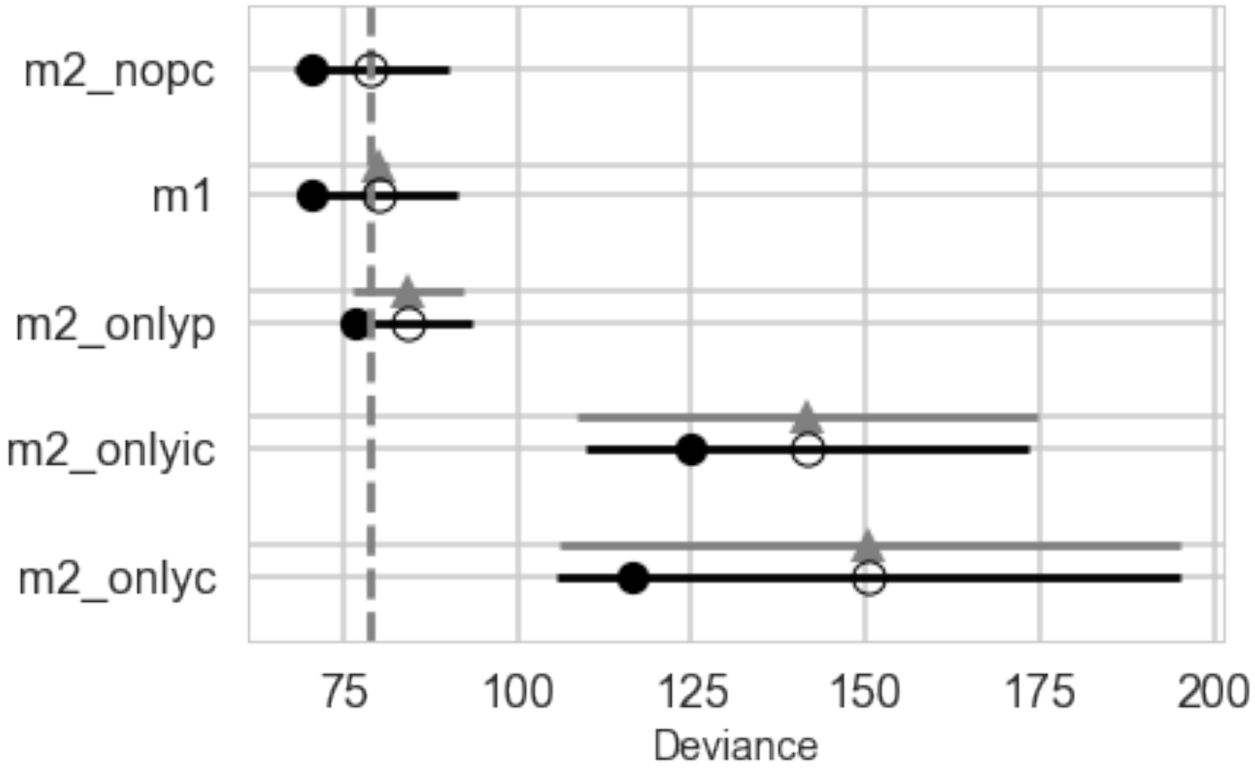
$$\sum_i Var_{post}[log(p(y_i|\theta))]$$

Definitions

- dWAIC is the difference between each WAIC and the lowest WAIC.
- SE is the standard error of the WAIC estimate.
- dSE is the standard error of the difference in WAIC between each model and the top-ranked model.

$$w_i = \frac{exp(-\frac{1}{2}dWAIC_i)}{\sum_j exp(-\frac{1}{2}dWAIC_j)}$$

read each weight as an estimated probability that each model will perform best on future data.



LOOCV

- Fit a model on N-1 data points, and use the Nth point as a validation point.
- the N-point and N-1 point posteriors are likely to be quite similar, use importance sampling. Fit the full posterior once. Then we have

$$w_s = \frac{p(\theta_s|y_{-i})}{p(\theta_s|y)} \propto \frac{1}{p(y_i|\theta_s, y_{-i})}$$

- the importance sampling weights can be unstablein the tails, pymc (`pm.loo`) fits a generalized pareto to the tail (largest 20% importance ratios) for each held out data point i (a MLE fit). Smooths out any large variations.

$$elpd_{loo} = \sum_i log(p(y_i|y_{-i})) = \sum_i log\left(\frac{\sum_s w_{is} p(y_i|\theta_s)}{\sum_s w_{is}}\right)$$

What should you use?

1. LOOCV and WAIC are fine. The former can be used for models not having the same likelihood, the latter can be used with models having the same likelihood.

2. WAIC is fast and computationally less intensive, so for same-likelihood models (especially nested models where you are really performing feature selection), it is the first line of attack

3. One does not always have to do model selection. Sometimes just do posterior predictive checks to see how the predictions are, and you might deem it fine.

4. For hierarchical models, WAIC is best for predictive performance within an existing cluster or group. Cross validation is best for new observations from new groups

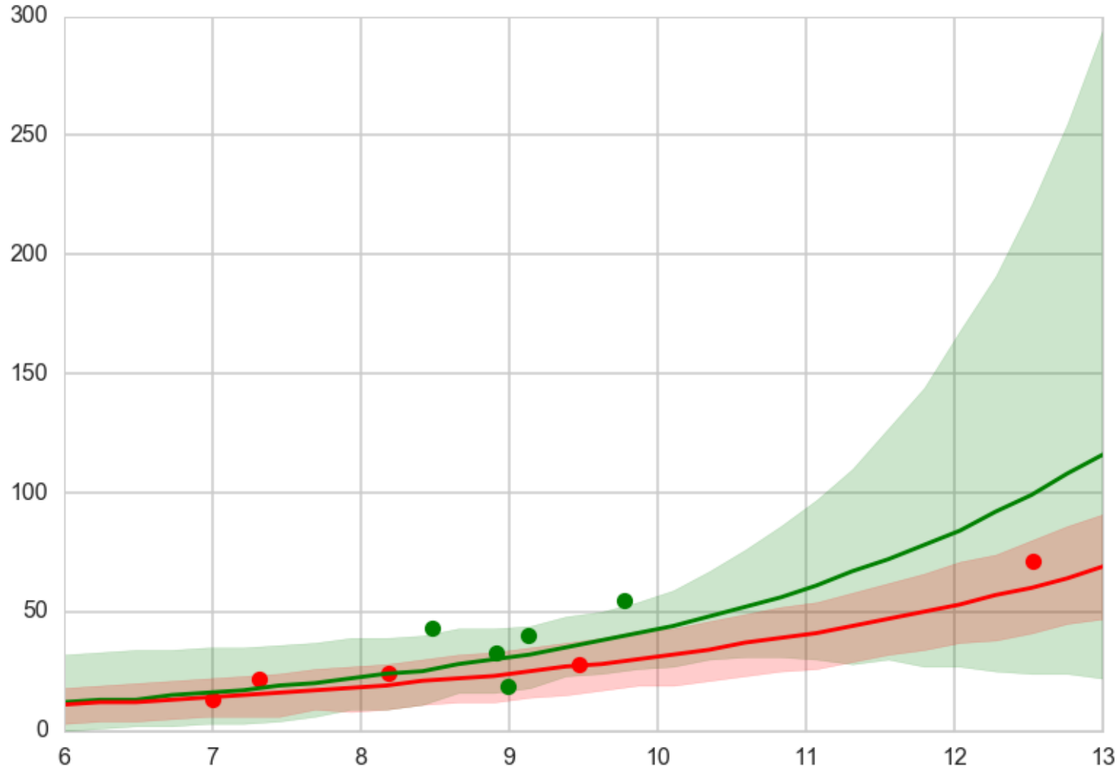
Bayesian Model Averaging

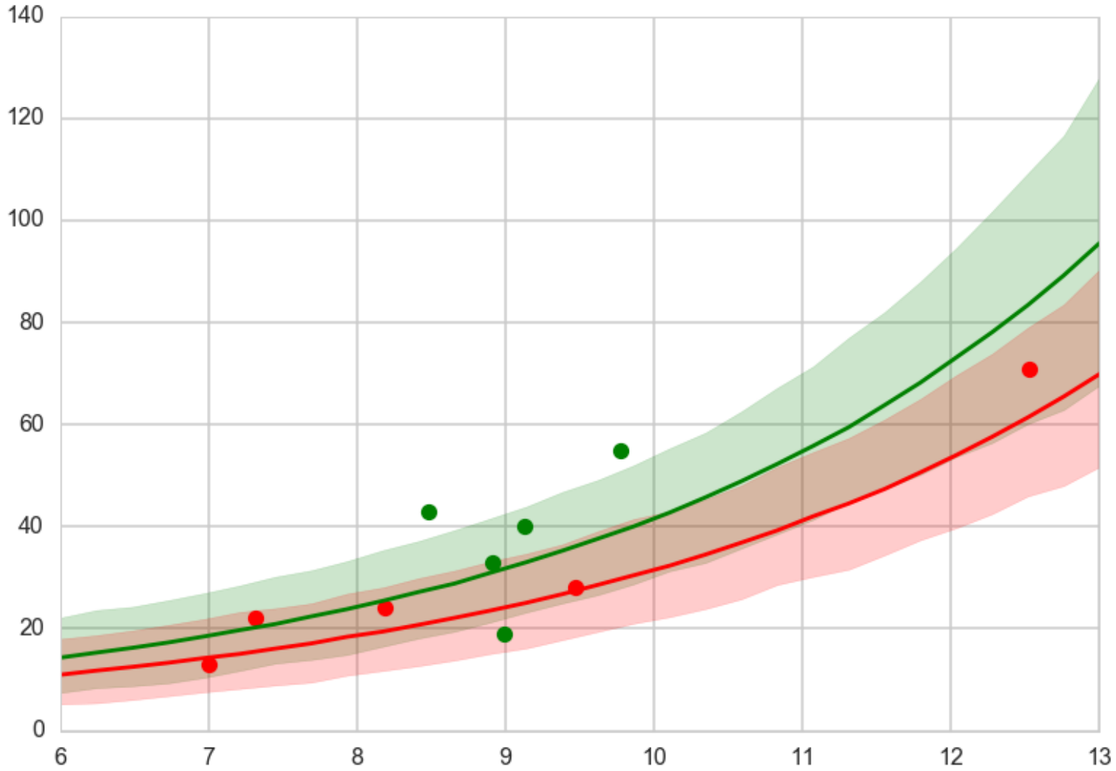
$$p_{BMA}(y^*|x^*, D) = \sum_k p(y^*|x^*, D, M_k)p(M_k|D)$$

where the averaging is with repect to weights $w_k = p(M_k|D)$, the posterior probabilities of the models M_k.

Use the weights from the WAIC

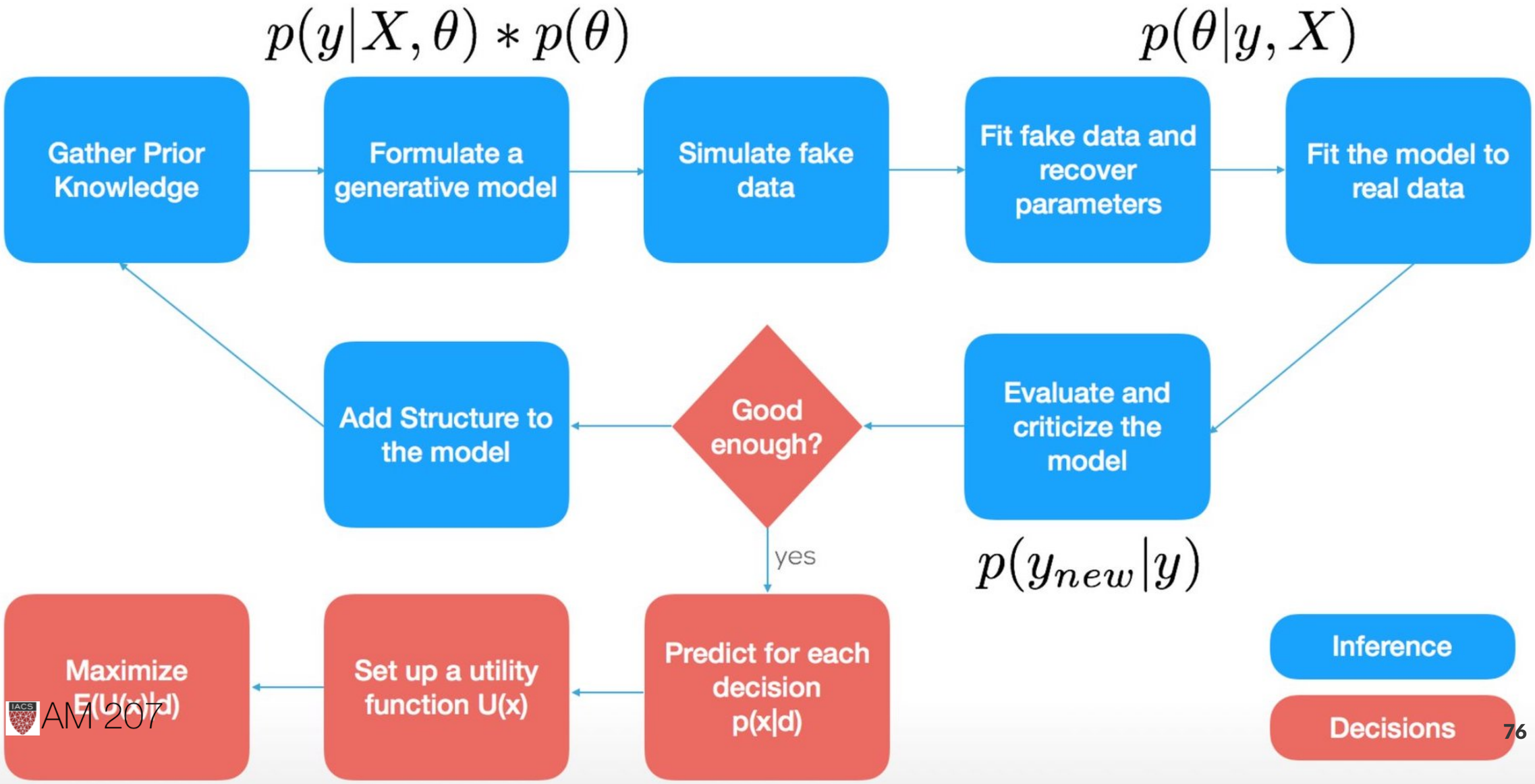
Counterfactual PP and ensembling via weights



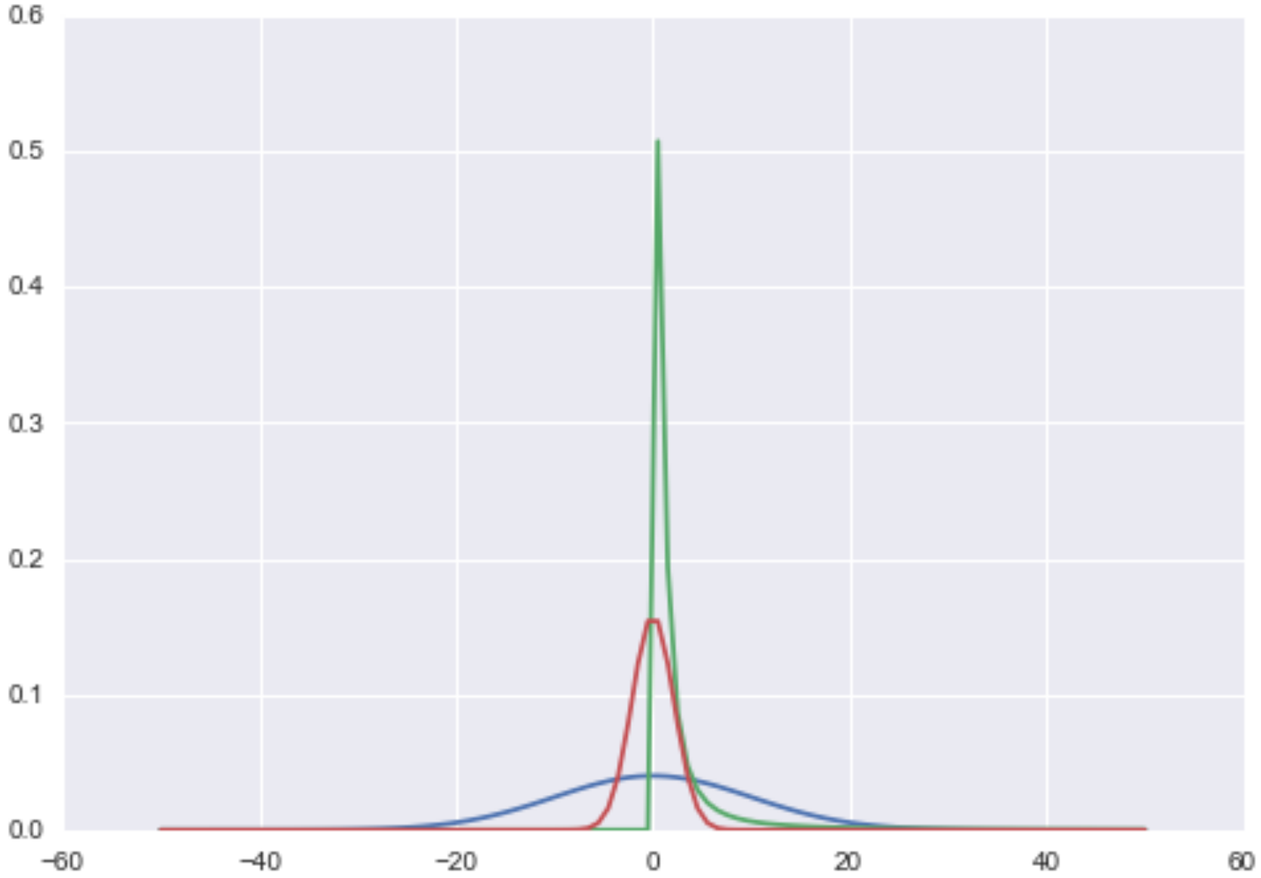


BAYESIAN WORKFLOW

Bayesian Workflow
(from @ericnovik)



PRIORS



- choose likelihoods with MAXENT
- choose priors as non-informative, e.g. uniform or Jeffreys
- better still: choose priors as weakly informative/regularizing
- helps with sampler performance
- sensible parameter space, should correspond to scales and units of process being modeled

Weakly informative or regularizing priors

- these are the priors we will concern ourselves most with
- restrict parameter ranges
- help samplers
- regularizing priors may have us using the data "twice" (hierachical models)
- see https://github.com/stan-dev/stan/wiki/Prior-Choice-Recommendations and Stan Manual

The Workflow (from Betancourt, and Savage)

Prior to Observation

1. Define Data and interesting statistics
2. Build Model
3. Analyze the joint, and its data marginal (prior predictive) and its summary statistics
4. fit posteriors to simulated data to calibrate
    - check sampler diagnostics, and correlate with simulated data
    - use rank statistics to evaluate prior-posterior consistency
    - check posterior behaviors and behaviors of decisions

Posterior to Observation

1. Fit the Observed Data and Evaluate the fit
    - check sampler diagnostics, poor performance means generative model not consistent with actual data
2. Analyze the Posterior Predictive Distribution
    - do posterior predictive checks, now comparing actual data with posterior-predictive simulations
    - consider expanding the model
3. Do model comparison
    - usually within a nested model, but you might want to apply a different modeling scheme, in which case use loo
    - you might want to ensemble instead

MIXTURES supervised formulation

$$Z \sim \text{Bernoulli}(\lambda)$$
$$X|z=0 \sim \mathcal{N}(\mu_0, \Sigma_0),\ X|z=1 \sim \mathcal{N}(\mu_1, \Sigma_1)$$

Full-data loglike: $l(x, z|\lambda, \mu_0, \mu_1, \Sigma) = -\sum_{i=1}^{m} \log((2\pi)^{n/2}|\Sigma|^{1/2})$

$$-\frac{1}{2}\sum_{i=1}^{m}\sum_{i=1}^{m}(x-\mu_{z_i})^T \Sigma^{-1}(x-\mu_{z_i}) + \sum_{i=1}^{m}[z_i \log\lambda + (1-z_i)\log(1-\lambda)]$$

Concrete Formulation of unsupervised learning

Estimate Parameters by $\mathbf{x}$-MLE:

$$
\begin{aligned}
l(x|\lambda, \mu, \Sigma) &= \sum_{i=1}^{m} \log p(x_i|\lambda, \mu, \Sigma) \\
&= \sum_{i=1}^{m} \log \sum_{z} p(x_i|z_i, \mu, \Sigma)\, p(z_i|\lambda)
\end{aligned}
$$

Not Solvable analytically! EM and Variational. Or do MCMC.

Supervised vs Unsupervised Learning

In **Supervised Learning**, Latent Variables $\mathbf{z}$ are observed.

In other words, we can write the full-data likelihood $p(\mathbf{x}, \mathbf{z})$

In **Unsupervised Learning**, Latent Variables $\mathbf{z}$ are hidden.

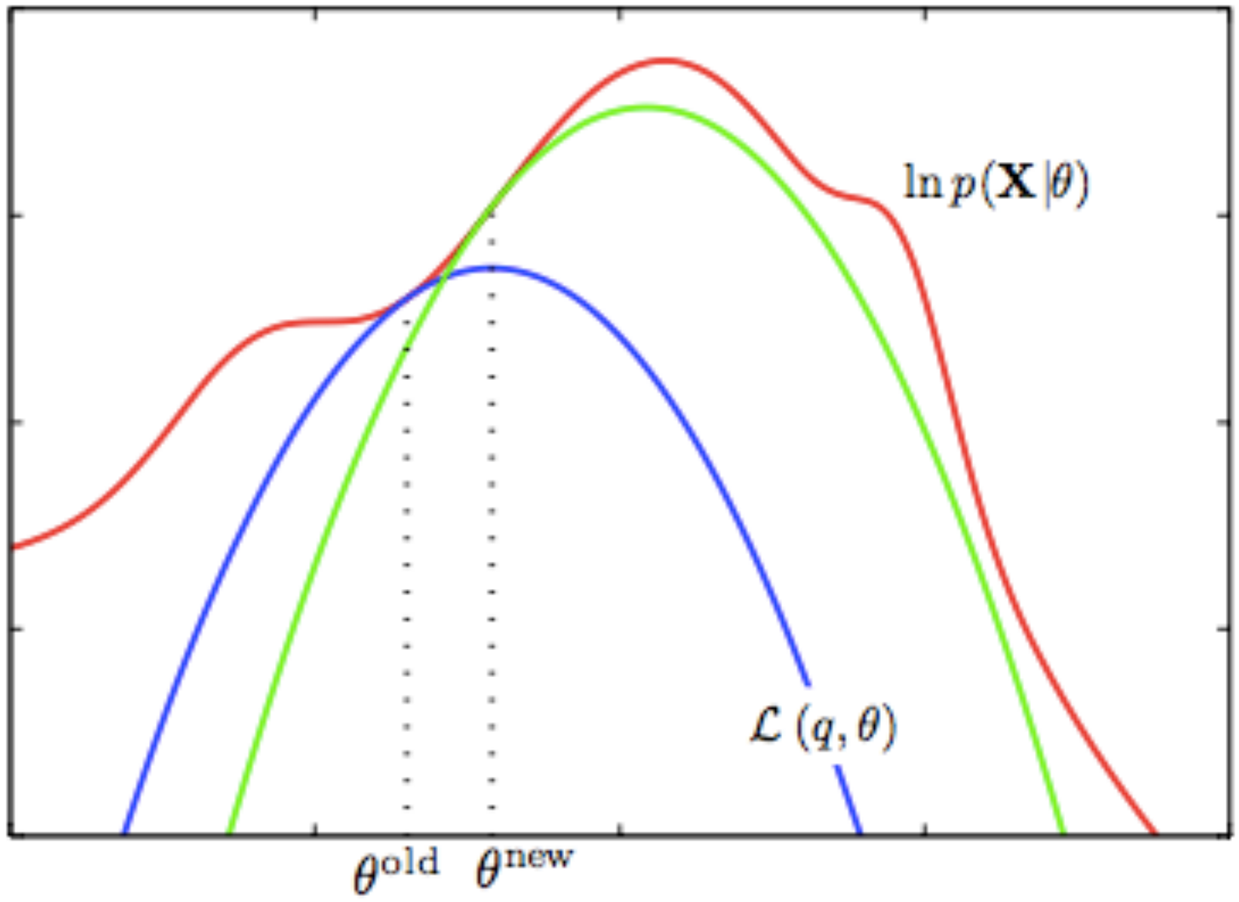
We can only write the observed data likelihood:

$$p(\mathbf{x}) = \sum_z p(\mathbf{x}, \mathbf{z}) = \sum_z p(\mathbf{z}) p(\mathbf{x}|\mathbf{z})$$

ELBO GREASE

EM

1. Start with $p(x|\theta)$(red curve), θ_{old}.
2. Until convergence:
    1. E-step: Evaluate $q(z, \theta_{old}) = p(z|x, \theta_{old})$ which gives rise to ELBO(θ): $\mathcal{L}(q(z, \theta_{old}), \theta)$(blue curve) whose value equals the value of $p(x|\theta)$ at θ_{old}.
    2. M-step: maximize ELBO (or Q func) wrt θ to get θ_{new}.
    3. Set $\theta_{old} = \theta_{new}$



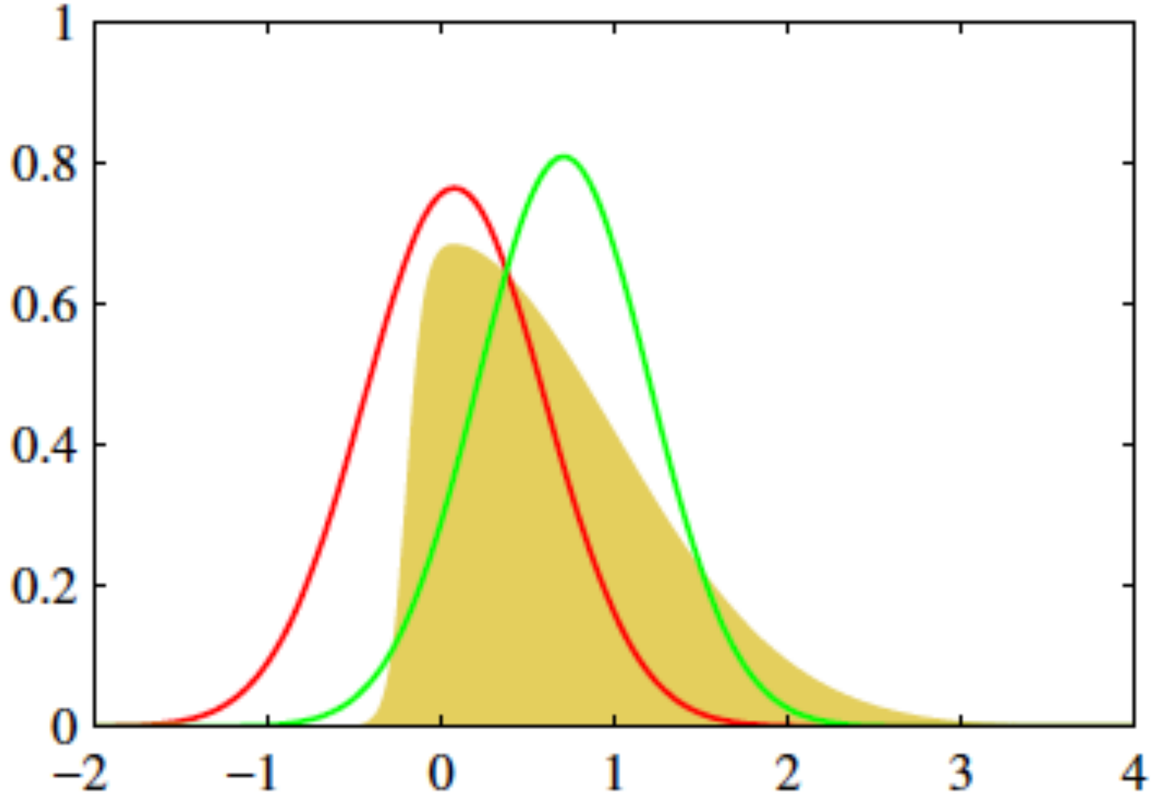
VARIATIONAL INFERENCE

Variational Inference Core Idea

z is now all parameters. Dont distinguish from θ.

Restricting to a family of approximate distributions D over z, find a member of that family that minimizes the KL divergence to the exact posterior. An optimization problem:

$$q^*(z) = \underset{q(z) \in D}{\arg\min} \quad KL(q(z)||p(z|x))$$



Mean Field: Find a q such that:

$KL + ELBO = log(p(x))$: KL minimized means ELBO maximized.

Choose a "mean-field" q such that:

$$q(z) = \prod_{j=1}^{m} q_j(z_j)$$

Each individual latent factor can take on any paramteric form corresponding to the latent variable.

ADVI

Core Idea:

- CAVI does not scale
- Use gradient based optimization, do it on less data
- do it automatically

What does ADVI do?

1. Transformation of latent parameters (**T** transform)
    - reparametrize mean field parameters to the real line
2. Standardization transform for posterior to push gradient inside expectation (**S** transform)
3. Monte-Carlo estimate of expectation
4. Hill-climb using automatic differentiation

Two ideas from Yao et. al.

- pareto shape parameter k from PSIS tells you goodness of fit (see here for @junpenglao pymc3 implementation, WIP). The idea comes from the process of smoothing in LOOCV estimation
- VSBC (variational simulation based callibration) : Extends calibration from Bayesian Workflow to variational case. pymc3 experimentation by @junpenglao here, WIP

Why use VB: Deep Generative Models

- simply not possible to do inference in large models
- inference in neural networks: understanding robustness, etc
- hierarchical neural networks (perhaps on exam)
- Mixture density networks: mixture parameters are fitted using ANNs
- extension to generative semisupervised learning
- variational autoencoders

Big Ideas

- learning is possible because there is a compressive manifold on which the data lives
- through SGD, HMC, etc we try tolearn about this manifold
- principled modeling can be done by combining known schemes such as poisson GLM with deep networks
- networks (which are just complex models) can be used at other places such as variational posteriors
- priors will regularise for us!

Interesting Times

- we progress by first predicting, and then understanding the robustness of our inference: posteriors and error bars
- MCMC/HMC, bayesian workflow, generative models, deep generative models and variational inference are at the cutting edge
- we have tried in this course to cover the basics and then be at this edge in places

What you have done and should do

- a lot of practice with lecture examples, labs, and homework
- been at the edge with your paper
- stay at the edge! Twitter is the place to be.
- follow folks like Andrew Gelman, Michael Betancourt, Jim Savage, Dan Simpson, Ian Goodfellow, Aki Vehtari, Dustin Tran, BayesGroup, Stephen Merity, Jeremy Howard, Roger Grosse, Ferenc Huszar, Alex D'Amour, Tom Wiecki, Colin Carrol, Tom Augsperger, Francios Chollet, Junpeng Lao, Richard McElreath

What Classes Should I take?

- Tamara Broderick at MIT, Bayesian Stats
- Denba's Decision Theory Course
- CS281 by Sasha Rush
- A host of stats courses: Statistical Nachine Learning, Bayesian Inference, glms
- Coursera Binge Watch: Daphne Koller's PGM course. The Bayesian part of Pedro Domingo's Machine Learning Course. Also Yaser Abu Mustafa's Learning from Data for Statistical Learning.

Books and Key Resources

- Our textbooks. Especially BDA for advanced stuff
- Murphy's Machine Learning a probabilistic perspective
- Bishops book (now online for free).
- Stan Manual and User guide. The User guide is priceless
- Stan-con Helsinki Videos

FIN