Lecture 22

Model Comparison and Ensembling

Decision Theory

Predictions (or actions based on predictions) are described by a utility or loss function, whose values can be computed given the observed data.

Point Predictions: squared loss

Sometimes we want to make point predictions. In this case a is a single number.

squared error loss/utility: $l(a, y^*) = (a - y^*)^2$

The optimal point prediction that minimizes the expected loss (negative expected utility):

$$\bar{l}\,(a) = \int dy^*\,(a - y^*)^2\,p(y^*|D, M),$$

is the posterior predictive mean:

$$\hat{a} = E_p[y^*].$$

The expected loss then becomes:

$$\bar{l}(\hat{a}) = \int dy^* \, (\hat{a} - y^*)^2 \, p(y^*|D, M) = \int dy^* \, (E_p[y^*] - y^*)^2 \, p(y^*|D, M) = Var_p[y^*]$$

Squared loss $\implies$ we dont care about skewness or kurtosis

This action $\hat{a}$ is called the **Bayes Action**.

Process

First define the distribution-averaged utility:

$$\bar{u}(a) = \int d\omega \, u(a, \omega) \, p(\omega|D)$$

We then find the a that maximizes this utility:

$$\hat{a} = \arg\max_a \bar{u}(a)$$

This action is called the **bayes action**.

The resulting maximized expected utility is given by:

$$\bar{u}(\hat{a}, p) = \bar{u}(\hat{a}) = \int d\omega \, u(\hat{a}, \omega) \, p(\omega|D),$$

sometimes referred to as the entropy function, and an associate **divergence** can be defined:

$$d(a, p) = \bar{u}(p, p) - \bar{u}(a, p)$$

Then one can think of minimizing $d(a, p)$ with respect to a to get $\hat{a}$, so that this discrepancy can be thought of as a loss function.

Log score: probabilistic prediction

Here we want to find a distribution a.

The utility is defined as:

$$u(a, y^*) = log\, a(y^*),$$

The expected utility then is

$$\bar{u}(a) = \int dy^*\, log(a(y^*))\, p(y^*|D, M).$$

The a that maximizes this utility is the predictive itself (in the bayesian context, the posterior predictive)!

$$\hat{a}(y^*) = p(y^*|D, M)$$

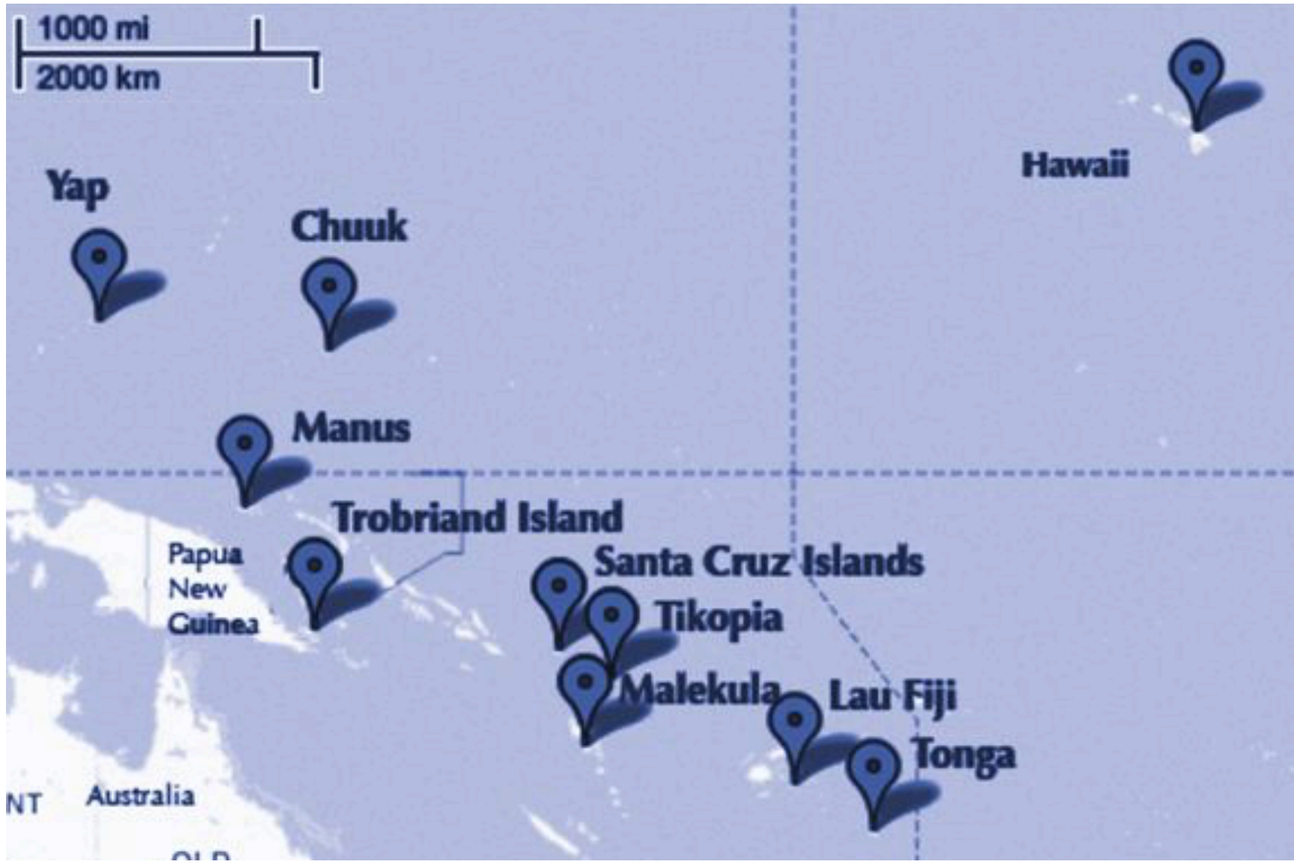
Maximized utility: $\bar{u}(a) = \int dy^* \, log(p(y^*|D, M)) \, p(y^*|D, M)$.

This is just the negative entropy of the predictive distribution, and the associated divergence is our old friend the KL-divergence.

Back to Poisson GLMs

From Mcelreath:

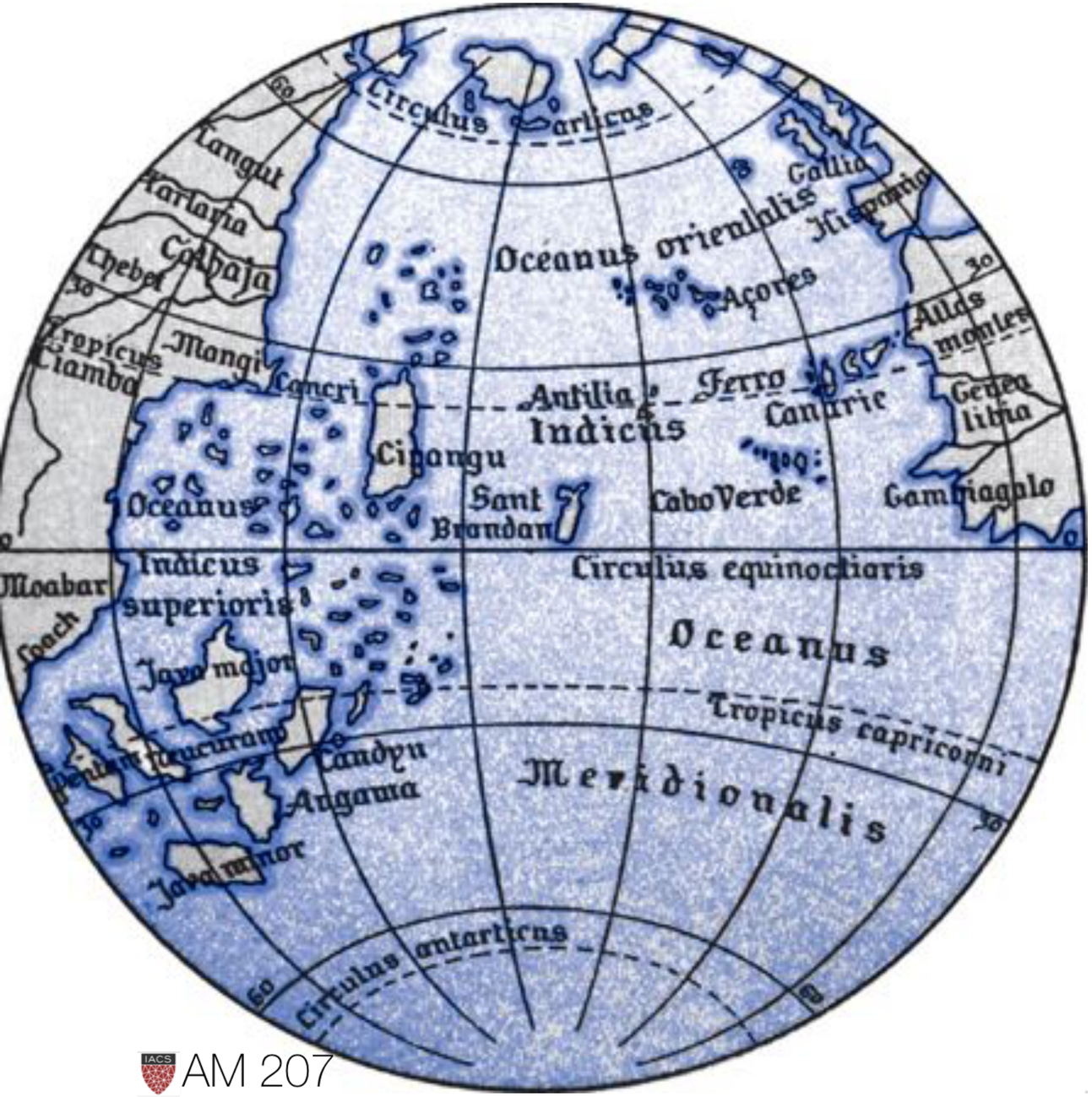
The island societies of Oceania provide a natural experiment in technological evolution. Different historical island populations possessed tool kits of different size. These kits include fish hooks, axes, boats, hand plows, and many other types of tools. A number of theories predict that larger populations 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)



Were the contacts really needed?

Let us compare models:

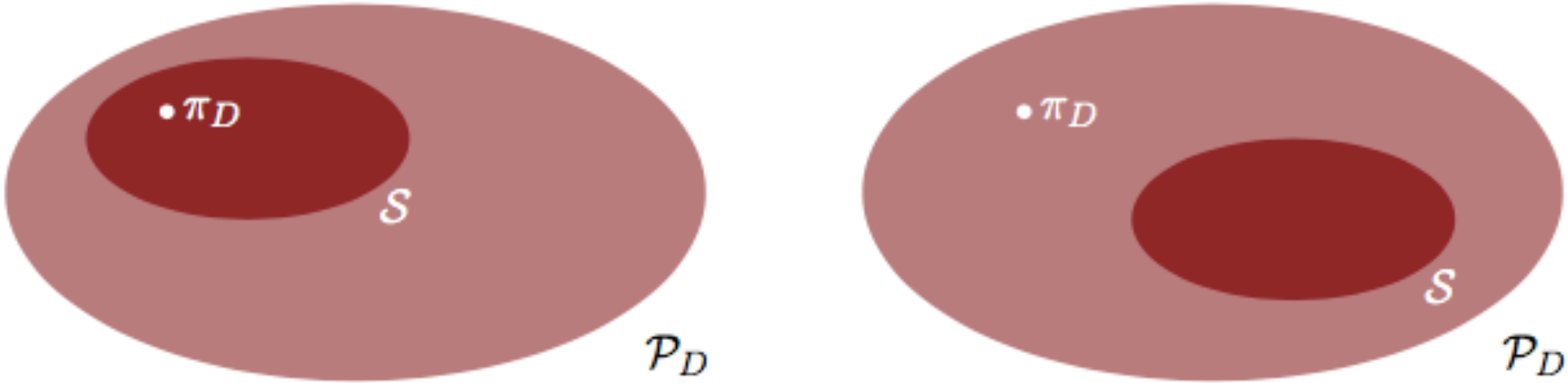
```
m2c_onlyic: loglam = alpha
m2c_onlyc: loglam = alpha +  betac*df.clevel
m2c_onlyp: loglam = alpha + betap*df.logpop_c
m2c_nopc: loglam = alpha + betap*df.logpop_c + betac*df.clevel
m1c: loglam = alpha + betap*df.logpop_c + betac*df.clevel + betapc*df.clevel*df.logpop_c
```



SMALL World vs BIG World

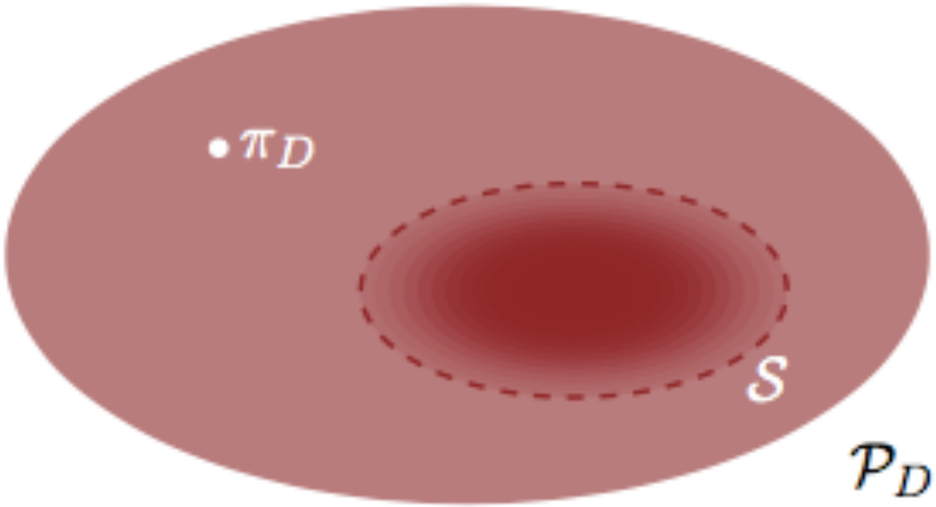
- *Small World* answers the question: given a model class (i.e. a Hypothesis space, whats the best model in it). It involves parameters. Its model checking.
- *BIG World* compares model spaces. Its model comparison with or without "hyperparameters".

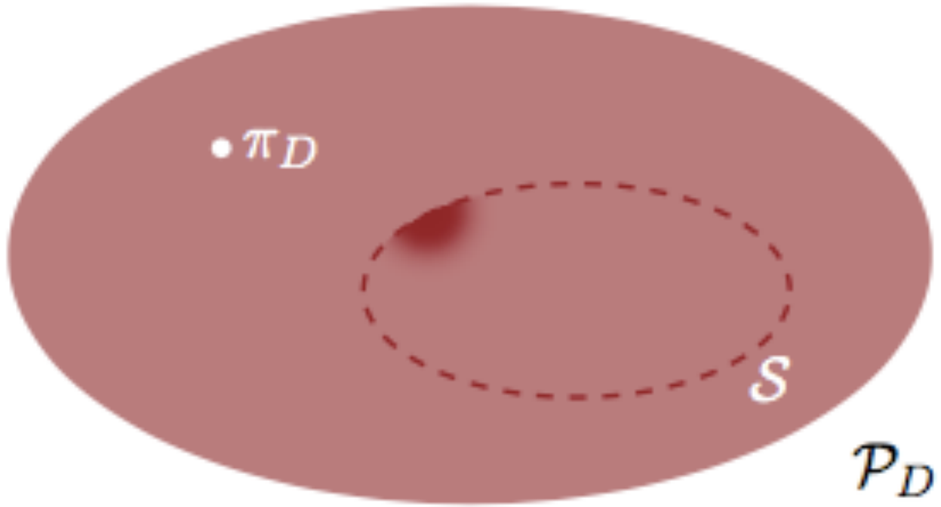
Bayesian Inference works in the small world



(some times includes the true generating process p_{tb} or p or $\pi_{\mathcal{D}}$)

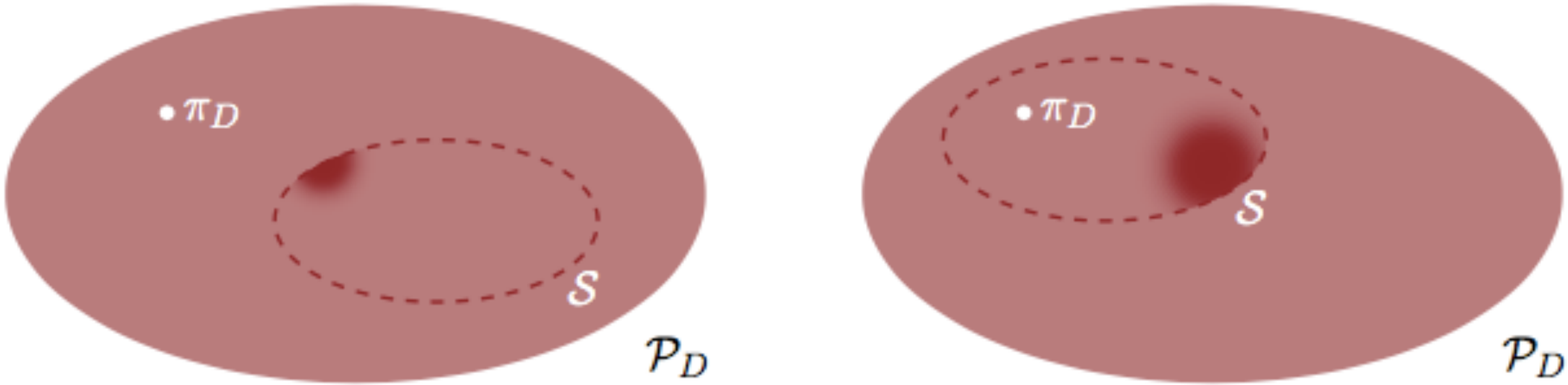
Inference in the small world





we go from prior to posterior

Bias and Variance



Overfitting can occur even if the small world includes the true data generating process p_{tb}.

How to compare against p_{tb}

In model comparison scenario we might use the "true" distribution:

$$\bar{u}_t(\hat{a}) = \int dy^* u(\hat{a}, y^*) p_{tb}(y^*)$$

Notice that we use $u(\hat{a}, y^*)$. The $\hat{a}$ has already been found by optimizing over our posterior predictive.

But we dont know p_{tb}. Does this remind you of something...?

True-belief distribution

- the "p" we used in KL-divergence formulae eons ago
- that is, the known or empirically estimated true distribution, or
- model M_{tb} that has undergone posterior predictive checks and is very expressive, a model we can use as a reference model.
- often non-parametric or found via bayesian model averaging.
- if the true generating process is outside the hypothesis set of the models you are using, true belief model never = true. This is called misfit or bias.

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}.

Utility is maximized with respect to some model $M_k \in \mathcal{H}$ whereas the average of the utility is computed with respect to either the true belief distribution.

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

where 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.

The MSE or R^2 in regression problems...

For the squared loss the first step gives us $\hat{a}_k = E_{p(y^*|D,M_k)}[y^*]$.

Then:

$$\bar{l}(\hat{a_k}) = \int dy^* \, (\hat{a}_k - y^*)^2 \, p(y^*|D, M_{tb})$$

$$= \int dy^* \, (E_{p_k}[y^*] - y^*)^2 \, p(y^*|D, M_{tb}) = Var_{p_{tb}}[y^*] + (E_{p_{tb}}[y^*] - E_{p_k}[y^*])^2$$

We have bias if M_{tb} is not in our Hypothesis set $\mathcal{H}$.

So far...

- Decisions are made on some validation or test set
- ensures that we dont use data twice
- while we are not picking hyper-parameters in the bayesian scenario
- we are still maximizing utility/minimizing risk
- and furthermore choosing a "best" model according to this utility

Information criteria

- we dont want to go out-of-sample
- use information criteria to decide between models

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

Use **law or large numbers** to replace the true distribution by empirical estimate

Deviance

$$D_{KL}(p,q) = E_p[log(p/q)] = \frac{1}{N}\sum_i (log(p_i) - log(q_i))$$

Deviance $D(q) = -\frac{N}{2} E_p[log(q)] = -2\sum_i log(q_i),$

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

Deviance of a predictive

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

We want to estimate the "true-belief" average of a predictive:

$$E_p[log(pred(y^*))]$$

where $pred(y^*)$ is the predictive for points y^* on the test set or future data.

Do it pointwise instead

Call the expected log predictive density at a "new" point:

$$elpd_i = E_p[log(pred(y_i^*)]$$

Then the "expected log pointwise predictive density" is

$$elppd = \sum_i E_p[log(pred(y_i^*))] = \sum_i elpd_i$$

What predictive distribution $pred$ do we use? We start from the frequentist scenario of using the likelihood at the MLE for the AIC, then move to using the likelihood at the posterior mean (a sort of plug in approximation) for the DIC, and finally to the fully Bayesian WAIC.

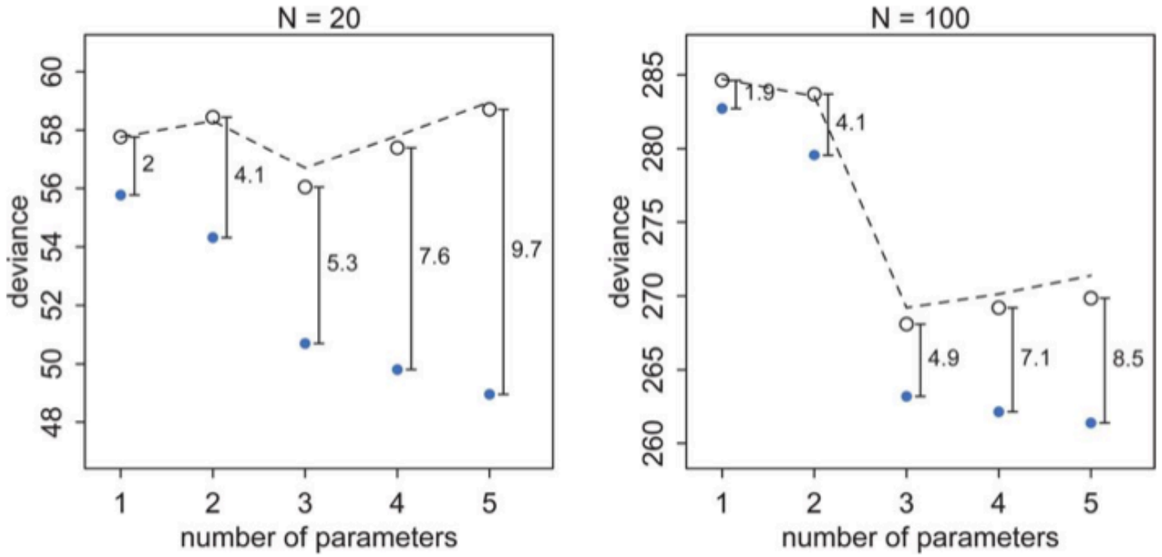
Specifically, in the first two cases, we are writing the predictive distribution conditioned on a point estimate from the posterior:

$$elpd_i = E_p[log(pred(y_i^* \mid \hat{\theta}))]$$

The game we will play in these first two cases is:

(1) Conditional on fixed θ, the full predictive splits into a product per point so the writing of elppd as a sum over pointwise elpd is exact
(2) However we dont know p_{tb} (or just p), so we use the empirical distribution on the training set
(3) this underestimates the test set deviance as we learnt in the case of the AIC, so we must apply a correction factor.

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

DIC

Uses the posterior distribution, calculable from MCMC, and assumes multivariate gaussian posterior distribution.

$$D_{train} = -2 * log(p(y|\theta_{postmean}), DIC = D_{train} + 2p_{DIC}$$ where

$$p_{DIC} = 2 * (log(p(y|\theta_{postmean}) - E_{post}[log(p(y|\theta)])$$ (by monte carlo)

alternative fomulation for p_D, guaranteed to be positive, is

$$p_{DIC} = 2 * Var_{post}[log(p(y|\theta_{postmean}))]$$

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 pp over new points y by product of marginals:

$$elpd_i = E_p[log(pp(y_i^*)]$$

$$elppd = \sum_i E_p[log(pp(y_i^*))] = \sum_i elpd_i$$

Game is to REPLACE

$$elppd = \sum_i E_p[log(pp(y_i^*))]$$ where y_i^* are new points

by the computed "log pointwise predictive density" (lppd) **in-sample**

$$lppd = log\left(\prod_j pp(y_j)\right) = \sum_j log\langle p(y_j|\theta)\rangle_{post} = \sum_j log\left(\frac{1}{S}\sum_{s\sim post} p(y_j|\theta_s)\right)$$

- As we know now, is that **the $lppd$ of observed data y is an overestimate of the $elppd$ for future data**.
- Hence the plan is to like to start with the $llpd$ and then apply some sort of bias correction to get a reasonable estimate of $elppd$.

This gives us the WAIC (Widely Applicable Information Criterion or Watanable-Akaike Information Criterion)

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

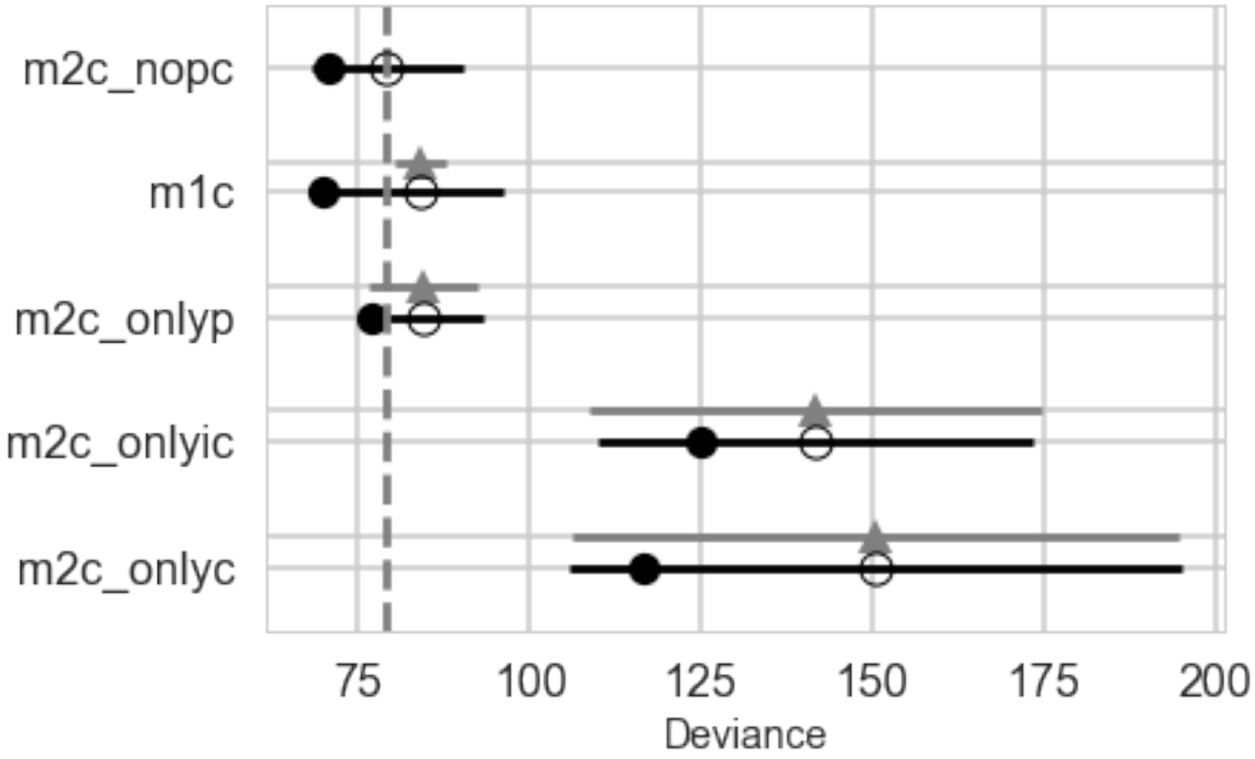
$$\sum_i Var_{post}[log(p(y_i|\theta))]$$

Oceanic tools

Lets use the WAIC to compare models

```
m2c_onlyic: loglam = alpha
m2c_onlyc: loglam = alpha +  betac*df.clevel
m2c_onlyp: loglam = alpha + betap*df.logpop_c
m2c_nopc: loglam = alpha + betap*df.logpop_c + betac*df.clevel
m1c: loglam = alpha + betap*df.logpop_c + betac*df.clevel + betapc*df.clevel*df.logpop_c
```

Centered



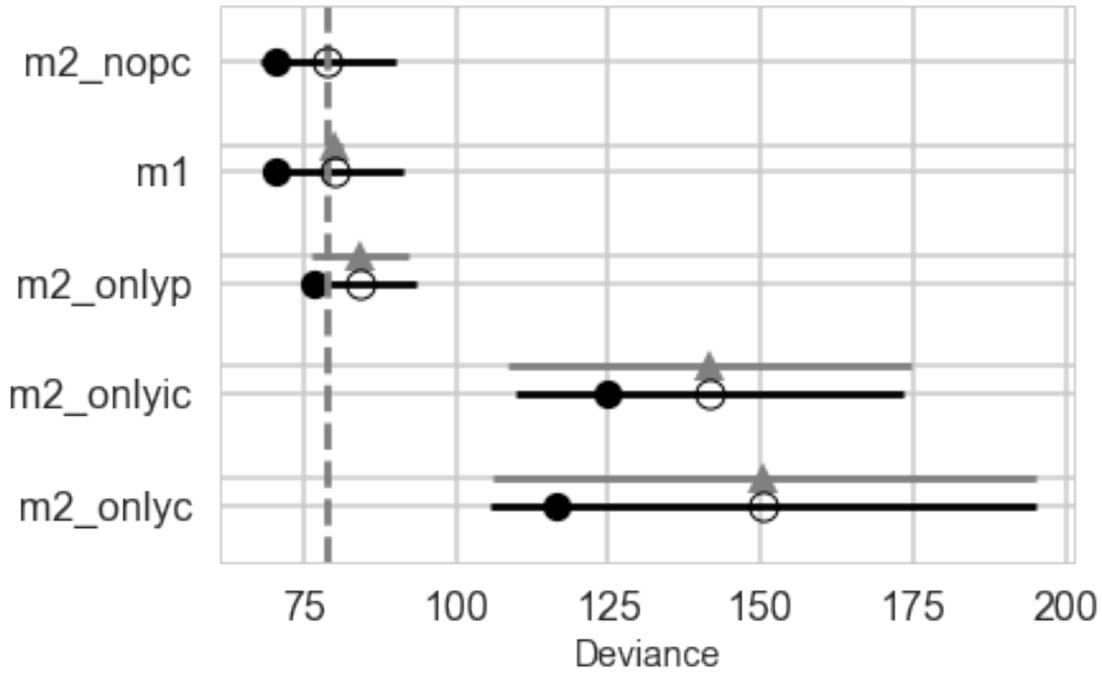
- dWAIC is the difference between each WAIC and the lowest WAIC.
- SE is the standard error of the WAIC estimate.
- dSE is the standard error of the difference in WAIC between each model and the top-ranked model.

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

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

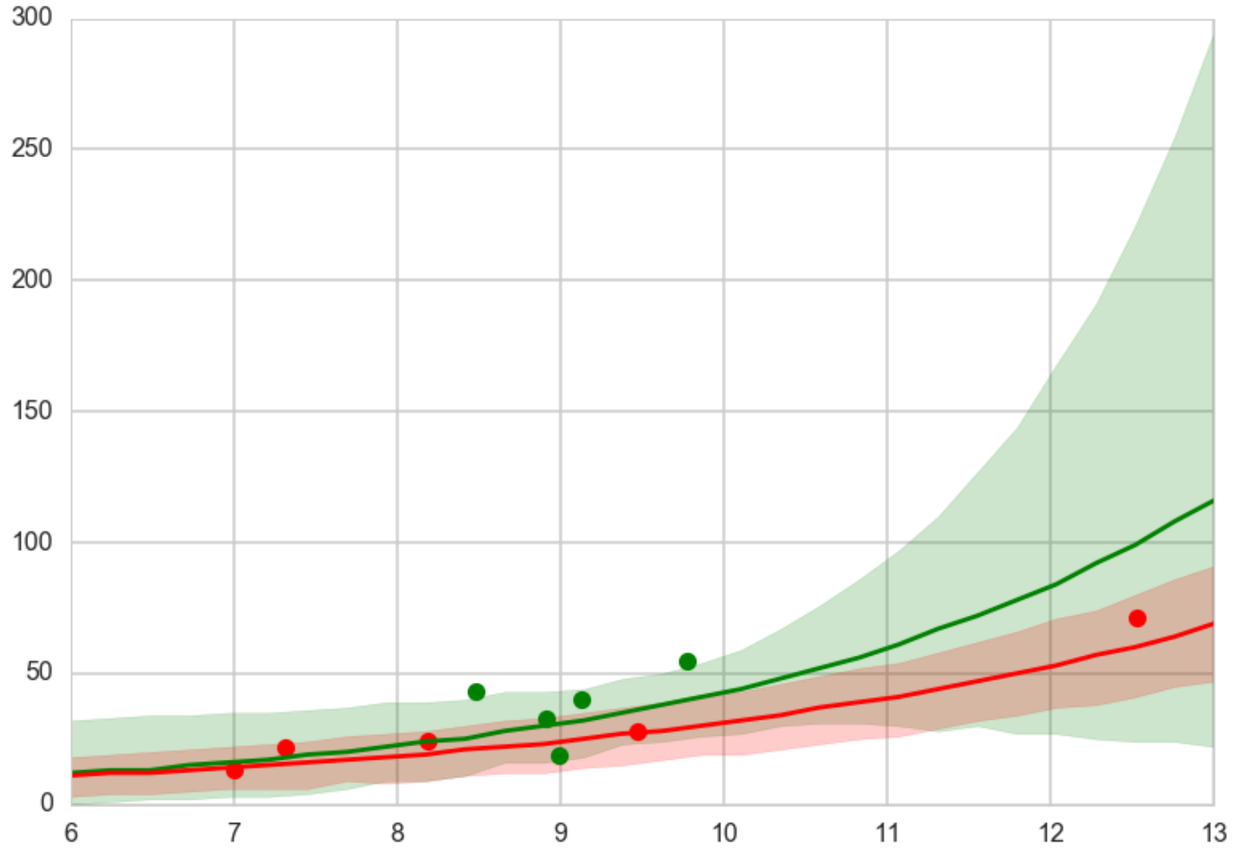
Uncentered

	WAIC	pWAIC	dWAIC	weight	SE	dSE	warning
name							
m2_nopc	79.1059	4.22647	0	0.61959	11.0612	0	1
m1	80.3046	5.03686	1.19871	0.340258	11.3985	0.571957	1
m2_onlyp	84.5787	3.84888	5.47276	0.0401523	8.98146	20.1717	1
m2_onlyic	141.327	8.10745	62.2212	1.90956e-14	31.6664	338.568	1
m2_onlyc	152.975	18.1559	73.8689	5.64512e-17	46.6488	678.014	1



interaction is overfit. centering decorrelates

Counterfactual Posterior predictive



Bayes Theorem in model space

$$p(M_k|D) \propto p(D|M_k)p(M_k)$$

where:

$$p(D|M_k) = \int d\theta_k \, p(y|\theta_k, M_k)p(\theta_k|M_k)$$

is the marginal likelihood under each model. Can use these "Bayes Factors" to compare but high sensitivity to prior.

Bayesian Model Averaging

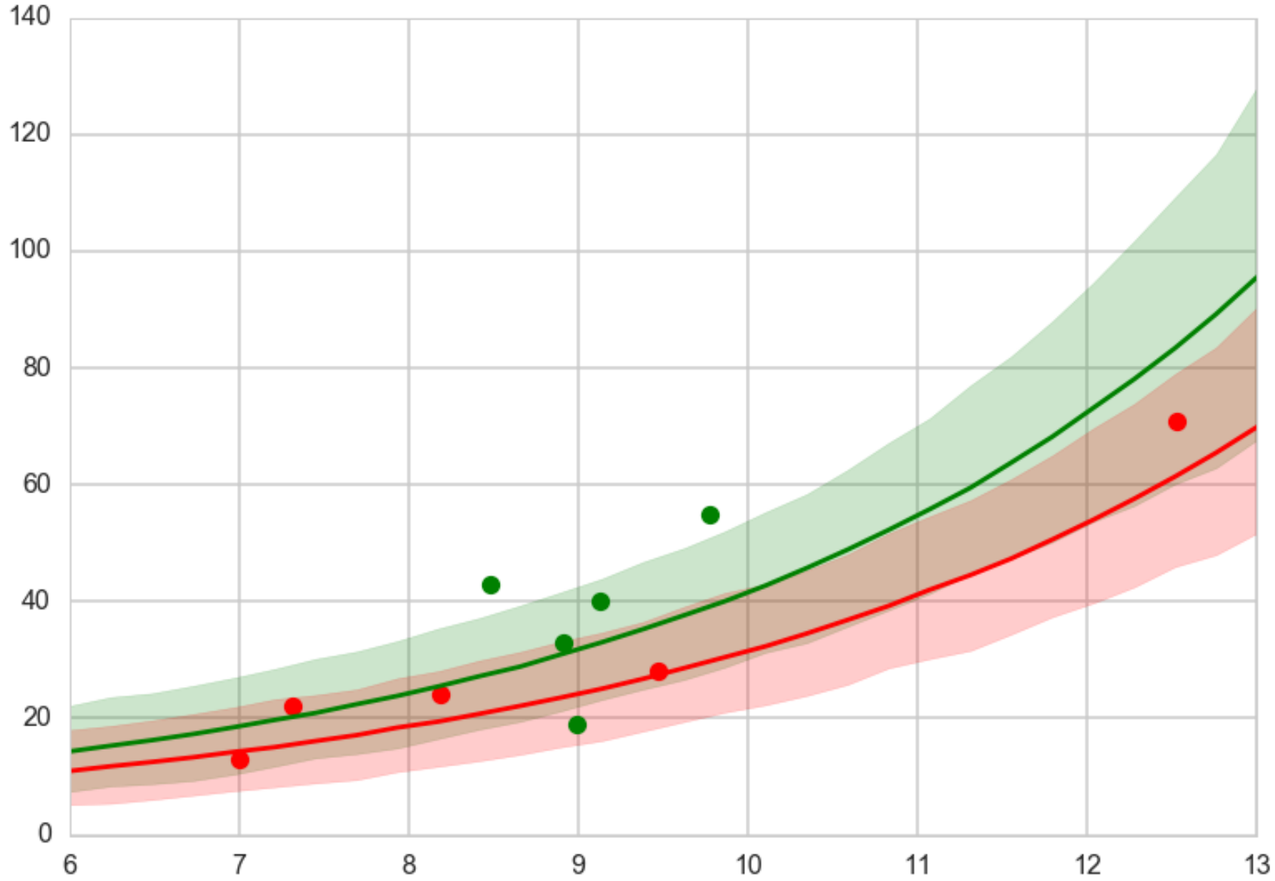
$$p_{BMA}(y^*|x^*, D) = \sum_k p(y^*|x^*, D, M_k)p(M_k|D)$$

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

We will use the "Akaike" weights from the WAIC. This is called `pseudo-BMA`

Ensembling

- use WAIC based akaike weights for top 3
- regularizes down the green band at high population by giving more weight to the no-interaction model.



- BMA is appropriate in the M-closed case, which is when the true generating process is one of the models
- what we will use here is to estimate weights by the WAIC, following McElreath (pseudo-BMA)
- But see Yao et. al. which claims log-score stacking is better. Implemented in pymc3

$$\max_{w} \frac{1}{n} \sum_{i=1}^{n} \log \sum_{k=1}^{K} w_k p(y_i|y_{-i}, M_k), \quad s.t. \quad w_k \geq 0, \quad \sum_{k=1}^{K} w_k = 1.$$

Pseudo BMA vs stacking

name	WAIC	pWAIC	dWAIC	weight	SE	dSE	warning
m2c_nopc	79.06	4.24	0	0.87	11.06	0	1
m1c	84.09	7.05	5.04	0.07	12.19	3.77	1
m2c_onlyp	84.43	3.75	5.37	0.06	8.94	7.93	1
m2c_onlyic	141.65	8.38	62.6	0	31.7	32.84	1
m2c_onlyc	150.44	16.94	71.38	0	44.67	44.44	1

name	WAIC	pWAIC	dWAIC	weight	SE	dSE	warning
m2c_nopc	79.06	4.24	0	0.76	11.06	0	1
m1c	84.09	7.05	5.04	0	12.19	3.77	1
m2c_onlyp	84.43	3.75	5.37	0.24	8.94	7.93	1
m2c_onlyic	141.65	8.38	62.6	0	31.7	32.84	1
m2c_onlyc	150.44	16.94	71.38	0	44.67	44.44	1

...it is tempting to use information criteria to compare models with different likelihood functions. Is a Gaussian or binomial better? Can't we just let WAIC sort it out?

Unfortunately, WAIC (or any other information criterion) cannot sort it out. The problem is that deviance is part normalizing constant. The constant affects the absolute magnitude of the deviance, but it doesn't affect fit to data.

— McElreath

How to handle non-nested models?

- cross-validation
- less data to fit so biased models
- we are not talking here about cross-validation to do hyperparameter optimization
- specifically we will use Leave-One-Out-Cross-Validation (LOOCV) with importance sampling

LOOCV

- The idea here is that you fit a model on N-1 data points, and use the Nth point as a validation point. Clearly this can be done in N ways.
- the N-point and N-1 point posteriors are likely to be quite similar, and one can sample one from the other by using importance sampling.

$$E_f[h] = \frac{\sum_s w_s h_s}{\sum_s w_s} \text{ where } w_s = f_s/g_s.$$

Fit the full posterior once. Then we have

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

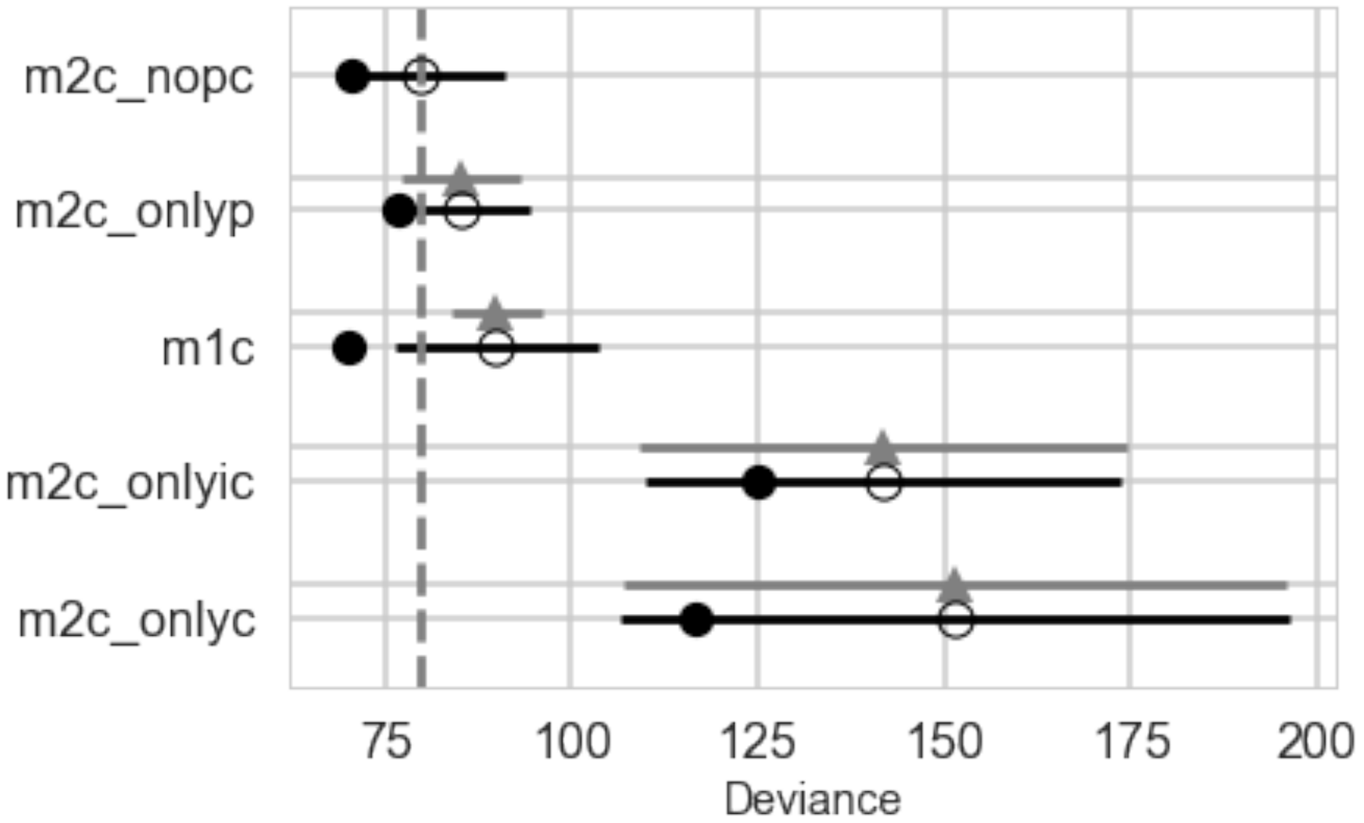
- the importance sampling weights can be unstable out in the tails.
- importance weights have a long right tail, pymc (`pm.loo`) fits a generalized pareto to the tail (largest 20% importance ratios) for each held out data point i (a MLE fit). This smooths out any large variations.

$$elpd_{loo} = \sum_i log(p(y_i|y_{-i}))$$

$$= \sum_i log\left(\frac{\sum_s w_{is} p(y_i|\theta_s)}{\sum_s w_{is}}\right)$$

over the training sample.

Oceanic tools LOOCV



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