Model Predictive Accuracy

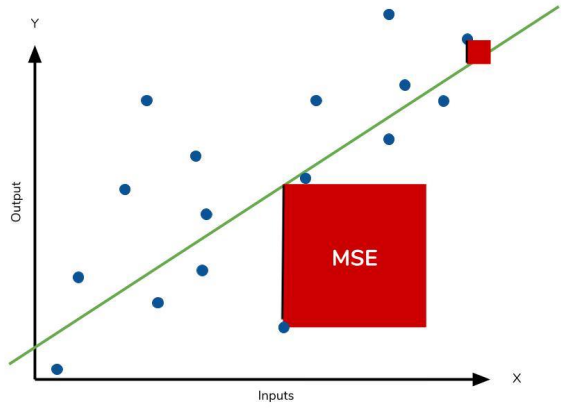
Bayesian Data Analysis

Steve Buyske

Mean-squared error

- You may have seen the *mean-squared error* (MSE) is often used as a measure of how well a model fits.
- In a regression context, we often write $\hat{y}_i$ for the predicted value of y_i, and then the mean-squared error is defined as

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2.$$



Mean-squared error cont.

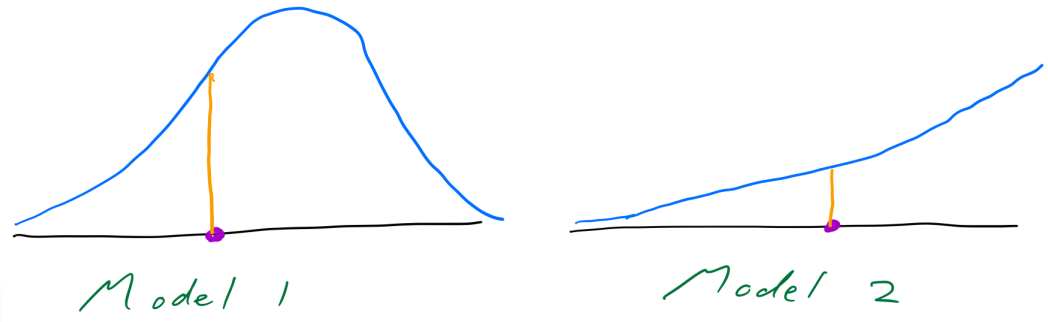
- For MSE, the smaller the better.
- The MSE is easy to compute.
- It is easy to interpret.
- Less appropriate if we are not using a normal model

The log score

- Suppose we have some data $\mathbf{y}$, as well as a particular model M_1 with parameters θ. Remember the likelihood is

$$p_1(\mathbf{y} \mid \theta).$$

- If this particular model and particular choice of θ fits the model well, then the probability at the particular observations will be high, and $p_1(\mathbf{y} \mid \theta)$ will be large.
- If there is a poor fit, then many observations will have low probability, and $p_1(\mathbf{y} \mid \theta)$ will be smaller.



The log score cont.

- For computational and theoretical reasons, we generally use the log of the likelihood, namely (dropping the subscript on p)

$$\log(p(\mathbf{y} \mid \theta)).$$

- This expression goes by many names, depending on context, but including the *log score* and the *log likelihood*.
- Not coincidentally, in the regression context it is also proportional to the MSE.

The log score cont.

- Because the $\log$ is what's called an increasing function (if $x_1 > x_2$, then $\log(x_1) > \log(x_2)$), we still have bigger values mean a better fit.
- Only comparisons among the log scores for different models matter, not the value of one log score for one model.
    - Change the data and the log score will change.
- Notice that the prior doesn't play a role—at this point we are talking only about the data model and the data.

The expected log predictive density for a new data point

- The expression $\log(p(\mathbf{y} \mid \theta))$ depends on both the data model and the parameters.
- The next step is to bring in the entire distribution for the parameters, namely the posterior distribution.
- Suppose we have a single new observation $\tilde{y}_i$. The fit for that new observation, taking into account the posterior distribution of θ, is

$$\log p_{\text{post}}(\tilde{y}_i) = \log \int p(\tilde{y}_i \mid \theta) p_{\text{post}}(\theta)\, d\theta.$$

- This is called the *log predictive density*.
- Remember those blue curves in the posterior predictive check? If every observation had the same covariates, then $p_{\text{post}}(\tilde{y})$ would represent the mean of those curves.

The expected log predictive density for a new data point cont.

- But there's more. We don't know what that new observation $\tilde{y}_i$ will be.
- Suppose the true (but unknown to us) distribution of $\tilde{y}_i$ has density q.
- The *expected log predictive density*, the elpd, is the expected value of the log predictive density for a new data point.

$$\text{elpd} = \int \log\big(p_{\text{post}}(\tilde{y})\big) q(\tilde{y})\, d\tilde{y}.$$

- Keep in mind that we don't actually know the probability density q.

The expected log pointwise predictive density for a new dataset

- Just to add to the fun, there's another acronym.
- The elpd is for a single future data point.
- If we are thinking of a future dataset of size n, quite possibly with different covariates for each point, then
- We define the expected log pointwise predictive density for a new dataset as

$$\text{elppd} = \sum_{i=1}^{n} \text{elpd}_{\tilde{y}_i}.$$

- That is, just add up the elpd's for each point in the new dataset.

A big problem with the elppd

- At first glance, the elppd is just want we want as a model comparison criterion.
    - It tells us the expected performance of any model against future data.
- The big problem is that it depends on q the true distribution of future data, which we don't know.

Approaches to estimating elppd

- There is no perfect way to estimate the elppd, but there are several approaches.
- We could just estimate the elpdd by calculating the equivalent in the existing data (details later).
  - This will overestimate the elpdd, since it is evaluated on the same data that was used to fit the model.
- We could start with estimate from the bullet point above and then correct it.
  - This is the basis for criteria known as AIC, DIC, and WAIC.
  - They (especially the WAIC) work fine on average, but can do badly in any individual scenario.
- We could use what it known as *cross-validation*.
  - Hold out some of the original data, fit the model on the remaining data, evaluate the fit on the held-out data, and repeat many times (details later).

The log pointwise predictive density

- Remember the log predictive density for an individual observation is (slide 7) $\log p_{\text{post}}(\tilde{y}_i)$.
- Given a full dataset of independent observations, the *log pointwise predictive density*, the lppd, is the log of the product of the individual likelihoods, namely

$$\text{lppd} = \log \prod_{i=1}^{n} p_{\text{post}}(\tilde{y}_i) = \sum_{i=1}^{n} \log p_{\text{post}}(\tilde{y}_i)$$

$$= \sum_{i=1}^{n} \log \int p(\tilde{y}_i \mid \theta) p_{\text{post}}(\theta)\, d\theta.$$

- We generally evaluate this by using the draws from the MCMC chains, to get

$$\text{computed lppd} = \sum_{i=1}^{n} \log \frac{1}{S} \sum_{s=1}^{S} p(\tilde{y}_i \mid \theta^s).$$

- The lppd uses all of the data to give an (over)estimate of the elppd.

Leave one out cross-validation

- To better predict how a model would do on future data, we might try to find a second dataset (sometimes called a *validation* or *replication* dataset).
  - Even this isn't perfect, since *a* future dataset may not represent *all* future datasets.
- Without a second dataset, you might choose to set aside part of the current data as a "future" dataset.
  - Seems disappointing not to be able to use that data in modeling.
  - Same problem as to whether the set-aside data fairly represents future data.
- The popular theoretical approach is called *leave one out cross-validation*, often written LOO-CV.

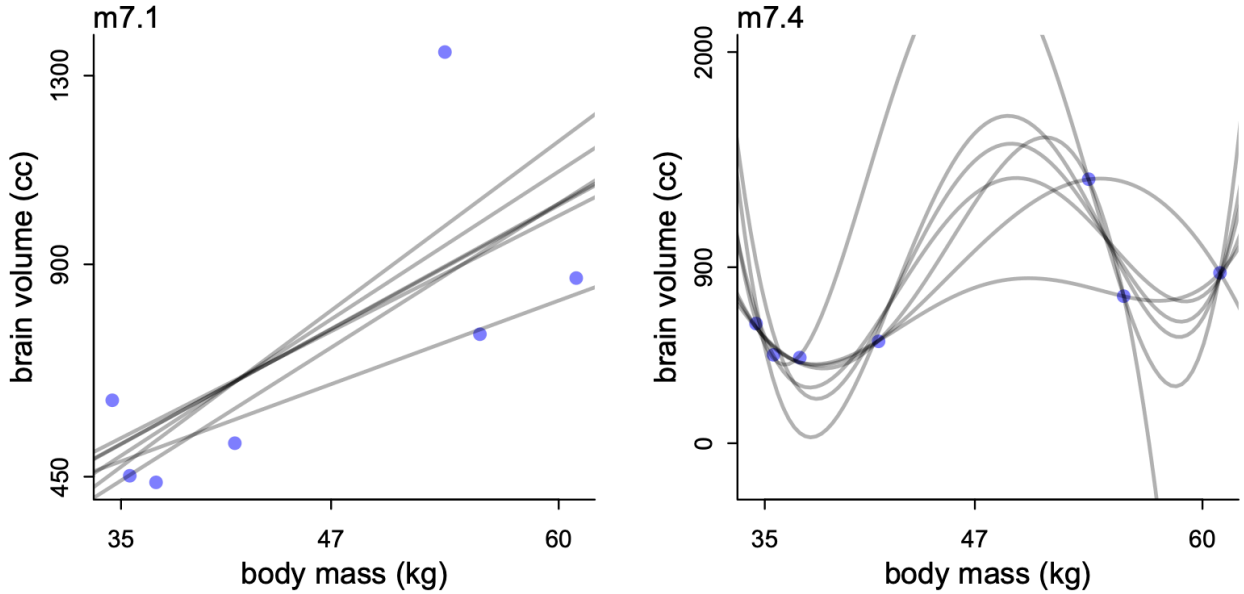


FIGURE 7.4. Underfitting and overfitting as under-sensitivity and over-sensitivity to sample. In both plots, a regression is fit to the seven sets of data made by dropping one row from the original data. Left: An underfit model is insensitive to the sample, changing little as individual points are dropped. Right: An overfit model is sensitive to the sample, changing dramatically as points are dropped.

Leave one out cross-validation cont.

- The LOO-CV approach:
  - Leave one observation out.
  - Refit the model on the remaining data
  - Calculate the log predictive density of the held-out observation, using the model fit from the remaining data.
    - Call it $\log p_{\text{post (-i)}}(y_i)$.
    - Calculate it using the draws from the MCMC process, with $\log \frac{1}{S} \sum_{s=1}^{S} p(y_i \mid \theta^s_{-i})$.
  - Repeat for every single observation
  - Estimate the elppd with the

$$\text{lppd}_{\text{loo-cv}} = \sum_{i=1}^{n} \log p_{\text{post (-i)}}(y_i).$$

Leave one out cross-validation cont.

- The $\text{lppd}_{\text{loo-cv}}$ is pretty much the best general-purpose method out there, but
- It is computationally intensive
  - If you have a sample size of n, you have to go through the whole model fitting process n times.

PSIS-LOO to the rescue

- It would be nice to have an approximation to the LOO-CV that is fast to calculate.
- There is! Here's the idea:
  - Every observation has an "importance" to determining the posterior (basically, the more unlikely the observation the higher the importance) that only needs to be calculated once.
  - The importance leads to weights, which are used to reweight the posterior (over observations and draws from the Markov chain) to estimate log predictive density for each the held-out observation
  - Actually, the weights are a bit unreliable, so they are smoothed out based on the knowledge that they should follow something called the Pareto distribution.

PSIS-LOO to the rescue

- Put it together, and you have the *Pareto Smoothed Importance Sampling Leave-One-Out Cross-Validation*, or PSIS LOO-CV, often just written PSIS.
- One advantage of the PSIS method is that it automatically detects that it may be giving a poor approximation to the LOO-CV whenever there are very large weights.
  - There is shape parameter k from the Pareto distribution—if that's above 0.7 then the PSIS LOO-CV is likely inaccurate.
- After calculating the PSIS LOO-CV, it is possible to find not just the differences in the estimated elppd's, but also a standard error of those differences.
  - I would keep under consideration any model with a difference less than 2.5 standard deviations from the best fitting model.

Final notes

- Statistical model comparison does not replace thinking.
- It has nothing to say about causality.
- Model comparison does clarify which models are supported by the evidence, or more to the point which are not.
- Because priors regularize parameter estimates, the Bayesian framework generally reduces overfitting.