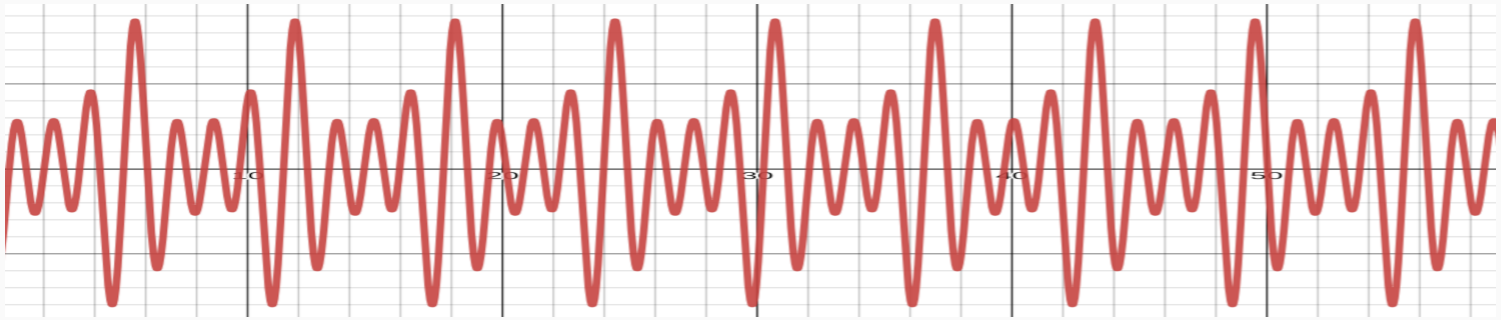
Model Selection



Prof. Fabrício Olivetti de França

Federal University of ABC

05 February, 2024

Model Selection

Model Selection

In previous lectures, we explored different SR algorithms and their many hyper-parameters.

The choice of the best hyper-parameters can be essential to generate a model that best describes the data generating process.

Choosing the best set is not trivial and there is no *one-fits-all*.

Model Selection

In the end, we have to make experiments with different SR algorithms and different hyper-parameters and select the best model somehow.

Even though we have a well-defined optimization criteria that we use during the search process, this criteria may be insufficient to capture the whole desiderata or may be biased towards the choice of training sample.

Also, in a Multi-objective setting, the algorithm may return a set of trade-off solutions. Should we pick the simplest one? The most accurate?

Model Selection

To make this choice, we first need to establish the goal of the experiment: statistical inference or prediction.

With **statistical inference** we want to understand the nature of data generating process, leading to a correct characterization of the sources of uncertainties and providing an explanation for the data.

The **prediction** objective consists of predicting unseen observations based on the current observations we have collected.

Hold-out set

A simple approach is to create a hold-out set, in short we split the data into a training set and a test set.

In this approach, we use the training set to search for or fit the model, while the test set is used to evaluate the final model.

The main idea is that, using the training set to evaluate the model holds a bias toward the search and optimization process. Having a separate set for evaluation answers the question “What is the expected performance of this model when applied to unseen data”.

Hold-out set

Notice that if we also want to determine the best hyper-parameters for the algorithm, we may split the training data even further as a training and validation sets.

In this scenario, we use the training set for the search and fitting, and the validation to evaluate whether that specific hyper-parameter is expected to return a good performing model.

Hold-out set

In GP-SR we can go even deeper and split the set once more to make sure that the fitness of the solution is unbiased.

These choices all depend on the data availability, since each splitting demands a large enough data set so that we have enough data available for the fitting process.

The ratio of split also depends on how many data points we have, a common setting for medium to large data is 70-20-10 for training, validation, and test.

As a rule-of-the-thumb for linear regression, it is desirable to have at least 10 data points for every parameter in the model. But, if possible, it is desirable to have 100 data points for each parameter.

Cross-Validation

Thinking of a classification problem, using a simple training-test split may generate:

3	1	4	0	Spam
5	3	0	0	Spam
1	1	0	2	Ham
5	3	0	0	Spam
1	1	0	2	Ham
5	3	0	0	Spam
1	1	0	2	Ham
5	3	0	0	Spam
1	1	0	2	Ham
5	3	0	0	Spam
5	3	0	0	Spam
5	3	0	0	Ham
1	1	0	2	Ham
1	1	0	2	Ham

We cannot guarantee that either the training or the test sets are representative.

Cross-Validation

If the training set is not representative, the model will be fitted with a bias towards the represented region.

If the test set is not representative, the choice of model will be of a biased model towards the represented region.

To avoid this situation we can use the k-fold cross-validation.

Cross-Validation

In k-**fold cross-validation** we split the data into k sets of (approximately) equal size and we run the fitting and evaluation procedure k times, each time using the combination of $k-1$ sets as the training data and the remaining split as the test set.

After repeating the experiment k times we apply an aggregation function into the performance measurements and choose the model that maximizes the aggregated performance. For example, we can calculate the average R^2 of the fitted models in k different settings.

Cross-Validation

$k = 4$, experiment 1:

3	1	4	0	Spam
5	3	0	0	Spam
1	1	0	2	Ham
5	3	0	0	Spam
1	1	0	2	Ham
5	3	0	0	Spam
1	1	0	2	Ham
5	3	0	0	Spam
1	1	0	2	Ham
5	3	0	0	Spam
5	3	0	0	Spam
5	3	0	0	Ham

Cross-Validation

$k = 4$, experiment 2:

3	1	4	0	Spam
5	3	0	0	Spam
1	1	0	2	Ham
5	3	0	0	Spam
1	1	0	2	Ham
5	3	0	0	Spam
1	1	0	2	Ham
5	3	0	0	Spam
1	1	0	2	Ham
5	3	0	0	Spam
5	3	0	0	Spam
5	3	0	0	Ham

Cross-Validation

$k = 4$, experiment 3:

3	1	4	0	Spam
5	3	0	0	Spam
1	1	0	2	Ham
5	3	0	0	Spam
1	1	0	2	Ham
5	3	0	0	Spam
1	1	0	2	Ham
5	3	0	0	Spam
1	1	0	2	Ham
5	3	0	0	Spam
5	3	0	0	Spam
5	3	0	0	Ham

Cross-Validation

$k = 4$, experiment 4:

3	1	4	0	Spam
5	3	0	0	Spam
1	1	0	2	Ham
5	3	0	0	Spam
1	1	0	2	Ham
5	3	0	0	Spam
1	1	0	2	Ham
5	3	0	0	Spam
1	1	0	2	Ham
5	3	0	0	Spam
5	3	0	0	Spam
5	3	0	0	Ham

Cross-validation

Some special cases:

- $k = 1$ there is no split
- $k = 2$ we split the data in half and performs two experiments
- $k = n$ we split the data in such a way that we get only a single test example for each experiment.

Leave-one-out

When $k = n$, the same size as the number of samples, we have the Leave-one-out technique.

This provides a much less biased estimation of the performance of the model on unseen observations.

But, if n is large, the runtime may be prohibitive (chiefly in populational SR), but it can be reasonable for smaller datasets.

Creating the final model with Cross-Validation

Once you choose the model using cross-validation technique, it is common to refit the model using the entire data to generate a better fit.

An alternative approach is to create a model as the ensemble of the k fitted models where the prediction is the average of the predictions of each model.

Information Theoretic Criterion

Akaike’s Information Criterion

Akaike Information Criterion (AIC) was created by Hirotsugu Akaike to measure the relative amount of information lost when choosing a model.

As already stated, we do not expect that the model will describe the data exactly, since we have many possible sources of uncertainties.

This quantity is based on the likelihood of the model and the number of adjustable parameters.

Akaike's Information Criterion

Given the maximum likelihood estimate $\hat{\mathcal{L}}$ of a model with k parameters, AIC is calculated as:

$$AIC = 2k - 2\ln\hat{\mathcal{L}}$$

Akaike's Information Criterion

Since we've been working with the negative log-likelihood, assuming $\hat{\text{nll}}$ as the minimum estimate of the nll:

$$AIC = 2(k + \hat{\text{nll}})$$

This is a minimization criteria, the smaller the number of parameters or the nll, the better. Intuitively, it will prefer more accurate models if they are not overparameterized.

Akaike's Information Criterion

This criteria estimates the Kullback-Lieber divergence between the model and the true generator function (if that was ever known). But, if the number of observations n is small, it requires some correction:

$$AICc = AIC + \frac{2k^2 + 2k}{n - k - 1}$$

This correction is valid if the model is univariate and linear following a normal distribution. Adapting to other scenarios is difficult.

Bayesian Information Criterion

Gideon E. Schwarz proposed the **Bayesian Information Criterion (BIC)** including a penalization for the number of observations:

$$BIC = k \ln n - 2 \ln \hat{\mathcal{L}}$$

$$BIC = k \ln n - 2\hat{\mathrm{nll}}$$

Bayesian Information Criterion

Compared to AIC, BIC will penalize the number of parameters weightd by the number of observations. The more observations, it will prefer even less parameters.

This comes in hand with the desiderata for SR in which we want a sufficient number of parameters to fit our data but not too many.

Like with AIC, this is a good approximation provided we have a large enough dataset.

Minimum Description Length Principle

As mentioned in the previous lectures, a regression model is a summarization of our data. When proposing a model for inference, we want it to accuretaly while being the most compact.

Viewing this task as a data compression, we want to find a compression model that can recover the that as best as possible but using the least information possible.

This principle is capture by the **Minimum Description Length (MDL)**.

Minimum Description Length Principle

A recent paper[1] proposed how to calculate MDL for SR. Their idea was that, given the data D and the functional hypothesis H, the *codelength* L of the data can be decomposed into:

$$L(D) = L(H) + L(D \mid H)$$

where $L(H)$ penalizes complex hypothesis and $L(D \mid H)$ is the accuracy of the hypothesis for that data.

[1] Bartlett, Deaglan J., Harry Desmond, and Pedro G. Ferreira. "Exhaustive symbolic regression." IEEE Transactions on Evolutionary Computation (2023).

Minimum Description Length Principle

Using this separation we can describe both AIC and BIC:

$$
\begin{aligned}
L(H)_{AIC} &= 2k \\
L(D \mid H)_{AIC} &= 2\hat{\mathrm{nll}} \\
L(H)_{BIC} &= k \ln n \\
L(D \mid H)_{BIC} &= 2\hat{\mathrm{nll}}
\end{aligned}
$$

Minimum Description Length Principle

Looking at the extremes, if we have $L(H) = 0$, we possibly have $L(D \mid H) = L(D)$ returning a very large error. Likewise, a very small error will lead to a very high model complexity.

Minimum Description Length Principle

$L(D \mid H)$ for SR is the minimum estimate of the negative log-likelihood for the model H.

The description length of the model $L(H)$ was split into two terms: the functional part and the parameter part.

Minimum Description Length Principle - Functional complexity

The hypothesis H describes a compressed data D, in order to retrieve the data we must transmit H in order to *decompress* the information.

Assuming H is written as an expression tree with k nodes. Then, we need to transmit k blocks of information each one containing the information about the operator or operand being transmitted.

If we have n operators in our set, we can represent this information using $\ln(n)$ nats, thus requiring $k \ln(n)$ in total.

Observation: we are using nats since we are already calculating the natural log of the likelihood, but we could also use bits ($\log_2$) or dits ($\log_{10}$).

Minimum Description Length Principle - Parameter complexity

Although parameters is also a parte of the symbols set, we must also transmit their values. Assuming a parameter with value θ_i and a precision $1/\Delta_i$, we need $\ln(|\theta_i|/\Delta_i) + \ln(2)$ nats to transmit its value. The last term captures the sign of the value.

Additionally, if we allow integral constant values in the expression (e.g., exponents, or as part of the simplification), we must add $\ln(c)$ nats of information for every constant.

Minimum Description Length Principle - Parameter complexity

Setting the parameter representation to a lower precision (i.e., increasing Δ_i) will likely reduce the precision of the maximum likelihood estimate, since the optimal parameter values may not be representable.

After finding Δ_i that minimizes the impact in the loss function, we have a parameter complexity of:

$$\frac{1}{2}\ln\left(I_{ii}\right) + \ln(|\theta_i|) - \frac{1}{2}\ln(3)$$

where I_{ii} is the i-th diagonal of the fisher matrix.

Minimum Description Length Principle

So we have:

$$MDL = \hat{\text{nll}} + k \ln n - \frac{p}{2} \ln 3 + \sum_j \ln(c_j) + \sum_{i=1}^{p} \frac{1}{2} \ln\left(I_{ii}\right) + \ln(|\theta_i|)$$

Using Model Selection

We must be aware that there no model selection criteria should be faced as an absolute truth. In fact, they can diverge in which model is the best.

The choice of which criteria to use depends on the number of observations, parameters, uncertainties, and how much accuracy is more important than interpretability.

A better use of such measures is to select a set of probable models and validate using one of the observations depicted in the previous lecture.

Next lecture

- Model Simplification

Acknowledgments