dissertation_writeup_draft

Alassane Ndour

23/08/2019

1 Introduction and Objectives

1.1 background to the problem

- Importance of growth models
- Noisy environment - Systematic error
- Classification using Bayes factor - e.g. See for a dataset classified as linear which model seems to fit (frequentist) better which will be selected by BF and by how much while varying noise

1.2 Reasons for the choice of project

- Application in biology and economics
- Contribution to literature as unexplored method - combination of Bayes factor and Harris paper
- Study of noise to signal ratio in classification and different types of noise

1.3 Identification of the project's beneficiaries

- Commercial partner (probably not but mught get data from them)
- Literature as an empirical analysis of Bayesian classification of growth using different models

1.4 Objectives and metrics

- A Classification framework which should include :
  - A classification between different models
  - The "certainty" of classification - TBD how we can quantify this
  - An estimation of the parameters of the model - with the "certainty" of estimation
  - An identification of the systematic error

1.5 Broad methods and how they answer goal

- Curve fitting :
  - Fit a linear and a logistic and classify depending on the error. See how as you increase the variance of the error, the classification changes (currently doing)
- Bayesian approach :
  - Estimate the distribution of the parameters (we should get the "certainty" from here) of a Bayesian linear regression, sigmoid function and then add the algorithm set by Harris (his calculation was for a sigmoid. Might have to do it for a linear regression) and compare the models using Bayes factor.
  - Does the model that fits the most correspond to the correct functional form?

  - See if as you change error the systematic error is caught by the Harris algo and how the model selection varies

- Compare the two approches : how do they compare ? In terms of classification error rate for instance
- Furthermore, there have been interesting developments in combining Bayesian methods and cross validation as they are not mutually exclusive methods and can contribute to robust estimates. Such works include Bürkner et al. (2019) where the authors aim to improve upon leave-future-out cross-validation (LFO-CV) - an adaptation of leave-one-out cross-validation (LOO-CV) to timeseries - to reduce computation time.

2 Context (Literature)

- Ed Harris
- LOO-CV
- Bayesian books
- Bürkner et al. (2019)
- Claeskens and Lid Hjort (2010) (Model selection)
-

3 Data

The data used in this project was generated data. There are several practical and methodological reasons for doing so. First, methodogically, generating data makes sense. As advised by Kéry and Royle (2016) as this offers a ideal control environment under which parameters and hyper-parameters are known. Furthermore, in growth cell literature, from which this project stems (e.g. Harris et al. (2016)) synthetic data is standard practice. Second, data from the commercial partner that was meant was to be analysed here was unavailable due to legal restrictions and no open source equivalents were found. As the synthetic data is at the heart of the analysis this section will describe in greater detail the data meant to be mimiced and the process/tools used to do so.

The type of data meant to be mimiced in this project is the similar to a cell counting process. The context posed by the commercial partner was the following : At any time `t` we must be able to estimate the number of a given cells that we wish to count. We have knowledge of the growth function that the cells take (i.e. `f(t)`). Now we know that introducing an agent in our cell sample alters the growth path to another growth process (`g(t)`). Given this we should be able to obtain the number of cells for any given time if we have knowledge of the presence of the agent. However if we do not know if the agent has been introduced in the sample then we must select whether we estimate the numbers of cells using `f(t)` or `g(t)`. We can apply a model selection in this case between the two growth process. Additionally, the counting process is subejct to a large amount of noise. Therefore, finding the ideal model selection method under noisy conditions describes problem to solve. In this case, the ideal model selection would favour a growth function that is able to find the true growth process along with the parameters associated to it. Furthermore, it is interesting for the researcher to be able to jauge the uncertainty surrounding the selected model and the parameters

Within this context, the data generated was meant to mimic a growth process through time. `x` represents the time through which the count increases. To bound the problem the count was normalized (min-max normalization) :

```
x belongs to [0, 1000] y belongs to [0 and 1]
```

The two growth functions used to generate the data were a simple linear function (1) and a logistic function (2) of the following forms :

`f(t) = alpha + beta * x (1)` where `alpha` is the intercept and `beta` the the coefficient of `x` and :

`g(t) = L/(1 +  e^(-k(x - x0))) (2)` where `L` describes the maximum value the curve could take, `k` describes the growth rate of the logistic function and `x0` represents the sigmoid's midpoint.

For all these parameters, the following values were uniformly drawn : * alpha ~ U[0, 0.05] * beta ~ U[0,0.2] * L ~ U[0.9, 1.1] * x0 ~ U[1/4 * max(x) , 3/4 * max(x)] * k ~ U[0.5, 2]

In order to simulate the noise in the problem and analyze it in a coherent manner different levels of additive Gaussian errors were introduced. The Gaussian errors all had a mean of 0 and a a variance `sigma` ranging from 0.1 to 1 by intervals of 0.1. In the following discussion we refer to each of these noise levels as noise buckets. Each noise bucket was comprised of 1000 synthetic datasets. To generate the data, custom numpy based functions were used. To understand the robustness of any of the selection processes used, we add to the generated data a drift which distorts the growth functional [TO DO]. From the synthtic data we record the x and y values (which are reffered to as the datasets) as well as the label (i.e. "linear" or "logistic"), the set of corresponding parameters and its corresponding noise bucket. A subset of the data used is presented in figure .. to provide clarity on the data used.

```r
head(dplyr::tibble(c("dataset", "variance_bucket", "drift_line", "functional_form")))
```

```
## # A tibble: 4 x 1
##   `c("dataset", "variance_bucket", "drift_line", "functional_form")`
##   <chr>
## 1 dataset
## 2 variance_bucket
## 3 drift_line
## 4 functional_form
```

4 Methods

This section describes the different estimation strategies used for model selection. To compare how well the different methods performed, we look at how well the datasets are classified correclty as well as how confident in the classification. Furthermore, as the systematic error and the parameters are of interest, we also focus on the estimated parameter values for different noise levels (and our confidence of these values) as well of when possible thes estimation of the eror itself. Finally, the time/complexity required for the estimation is also an important aspect of the study.

4.1 Frequentist approach: curve fitting

In the literature that tackles similar problems, curve fitting is the standard approch as demonstrated by [FIND REFERENCE]. In this study we use the same approach whoch we shall discuss in detail here. Curve fitting refers to the process of obtaining a mathematical function that can approxite a data. There are many approches to solve such a problem but one of the most common ones is to solve the least square problem shown in formally shown equation … . Least square aims at minimizing the sum of the distances between the fitted curve and the data points. Here a noticable difference has to be noted between a linear functional form and a logistic one: The former represents an unbound problem whereas the logistic function is by contruction bounded. This implies that different algorithms must be used in order to solve apply curve fitting to the two functions.

- (i) Solving the linear least square problem : Solving the linear regression problem is straightforward and a commmon result. As such the details of solving it are not expanded on here. If needed, readers can refer to [add reference].
- (ii) Solving the bounded non-linear least square problem : Solving the logistic curve fitting : trust region reflective algorithm (breif details on the algo). Given bounds were 0 and 1 - this mases sense as the researcher would have an idea of these bounds.

Both (i) and (ii) were solved using python's scientific library scipy.

Once both all the datasets were fitted with a logistic and a linear regression, a classification method was required. Here the MSE (equation required) was the prime candidate : the functional form with the lowest mean-squared error for a dataset would determine the predicted label. The advantage of this method was that it was possible to evaluate the confidence of the classification by observing the difference of the MSEs

- Standard approch
- method used linear uses lm and
- algorithm and package used used

4.2 Bayesian model selection

- find how we would tackle systematic error in this case - Ed Harris ##

5 Results

6 Discussion

7 Evaluation, Reflections, and Conclusions