A assignment of statistics using C++.

The writing here about generic stats/ML functions in C++ should not be used as factual information, but as an expression of my opinions.

To obtain a pdf of this page, click here. Please note that the pdf is not a 1 to 1 representation of this page and is rendered using `pandoc` and `texlive`.

The code is located here. This requires GNU make and a G++ with C++11 support.

The output

```
LogReg:
Coeff: P_0:0.999877 P_1:-2.41086
Accuracy: 0.46748
Sensitivity: 1
Specificity: 0
Time taken: 3558 ms
Naive Bayes:
Coeff:
class:
    0.172131    0.416667
    0.22541 0.262821
    0.602459    0.320513
sex:
    0.159836    0.679487
    0.840164    0.320513
age_mean: mu_0:30.3914 mu_1:28.8077
age_var: S_0:14.3232 S_1:14.4622
apriori: Surv_0:0.61 Surv_1:0.39
Accuracy: 0.735772
Sensitivity: 0.626087
Specificity: 0.832061
Time taken: 0 ms
```

Analysis

While logistic regression seems to have failed in terms of accuracy, this is possibly due to the fact that it is only measuring sex in terms of survival. The more pressing matter with logistic regression is the time taken to train the weights on the model itself. Using reasonable compiler optimizations (-O2, a good default optimization flag that doesn't introduce undefined behavior to the extent -O3 does), logistic regression training took 3.5 seconds to train. This is significantly slower to calculate than Naive Bayes, which took (apparently) less than a millisecond. Naive Bayes, possibly due to it's higher amount of parameters accessed, was significantly more accurate and specific, but it did have some false negatives unaccounted for.

Generative vs Discriminative

Generative models are a type of model that doesn't have the whole dataset in memory (with it's params, or actually holding the data), but models the data approximately. Parameter wise, it doesn't require a lot of data to be held and trained for doing math, but the parameters themselves are hyper tuned to the data itself. The computations themselves to make the trained parts are (most likely) very costly to preform, so they're precomputed. However, this means that only the weights need to be stored in the program binary, and calculations only need those weights in order to actually compute the models.

Discriminative models have the whole dataset, or a sample of the dataset. They can use the dataset to extrapolate new points in the dataset by directly comparing the new points to the dataset itself. It can even add new points to the dataset on the fly without having to retrain the parameters of the model. This, again, requires that a large sample of the data must be embedded in the program itself which may take up significant space in the program.

Reproducible Research in Machine Learning

Reproducible Research as defined here is a resilient method that does not produce significant variation when repeatedly trained over the same data with high probability. This means that models made can be reproduced by other people/entities. If this isn't the case, then the new models produced may have unforeseen concsequences. This may be compounded by costs of the data and of the training process itself too. [src] Because of this, models may not be trusted, and may just be a fluke of training instead of actual technological progress. A primary way of fixing the problem is to seed the randomness and use more complex data for statistical methods. Seeded, randomized weights allow for consistent reproduction of training methods.