Article from

Predictive Analytics and Futurism

June 2017
Issue 15

Bayesian Inference in Machine Learning

By Denis Perevalov

As the amount of data keeps growing, machine learning is drawing interest from different fields. With more data, one could find patterns and potentially use them in forecasts and recommendations. Maximum likelihood estimations (MLEs) are the most widely used machine learning methods, which is due to their speed and scalability. However, when dealing with smaller amounts of data or when data is narrow in the longitudinal direction, Bayesian analysis is arguably a better approach. Not only can it make more precise predictions, but its confidence intervals of model parameters are more interpretable.

Machine learning can be defined as the process of learning a predictive model's parameters from data. For a full specification of a problem, one has to have three ingredients: data, a predictive model hypothesis with parameters θ and a specification of the likelihood of observations, given the model and a set of predictive variables:

$$L(y|\theta, X)$$

where y is a vector of observations and X is a matrix of predictors.

The task of machine learning is the following: Given a training set of data (y, X), make the **inference** or **best estimate** of θ. In MLE, the latter is the one that yields the highest total likelihood in the training set:

$$\hat{\theta}_{best} = \text{argmax} L(y|\theta, X)$$

In the Bayesian approach, instead of a single point estimate $\hat{\theta}_{best}$, we predict a probability distribution function (PDF) of θ. We use the famous Bayes formula:

$$P(\theta|y, X) = \frac{p(\theta)L(y|\theta, X)}{\int p(\theta`)L(y|\theta`, X)d\theta`}$$

$P(\theta|y, X)$ and $p(\theta)$ are called **posterior** and **prior** distributions of θ, respectively. The integral in the denominator is a normalization constant, which is usually not important because we are interested in relative comparisons of θ.

The main feature of Bayesian analysis is that there is no optimization involved—it is simply a calculation of the posterior. However, the calculation should be performed for every single point in the space of θ. This is obviously unfeasible. Thus, we have to rely on the approximation of the posterior using samples of θ. In lower dimensions of θ, it is possible to do random sampling for the posterior estimation. In higher dimensions, one has to use more sophisticated sampling techniques. These techniques do not sample the entire θ space, but only its most likely part, and they still deliver an unbiased posterior estimation. Finally, because there is no optimization involved, there is no **overfitting** problem in the Bayesian inference.[1]

The prior distribution $p(\theta)$ is an assumption for the θ distribution before inferring it from the training data. It could be **informative** or **noninformative**. People talk about informative prior to when there is a good understanding of the θ distribution, which usually comes from an inference from some other data prior to the current study and results in a relatively narrow $p(\theta)$. Noninformative prior is used when conducting the study for the first time and when there is a very vague understanding of the θ distribution, maybe in terms of wide ranges. In that case, very wide prior distributions are used, such as normal with very high variance or uniform distribution with high width. Posterior lies somewhere within the prior distribution and is usually much narrower than the latter.

One has to be careful when choosing the prior distribution. For example, if one chooses prior to be uniform [–1,1], then posterior will always be somewhere in this interval, no matter what data suggests. The "Frequentism and Bayesianism" blog post has other good examples where a poor choice of prior may significantly bias the posterior.[2]

The main advantage of the Bayesian approach is that its result is a much richer description of the possible values of the model parameters. Apart from prior, the MLE result is only a special statistic for the Bayesian result—it is approximately its mode. MLE describes a single point in the θ space that is most likely, whereas a Bayesian result provides an entire distribution. For example, one could immediately calculate a mean, variance and skewness. In the case when one is interested in the expected value of θ, the mean is a more appropriate statistic than the mode, especially for highly skewed θ posteriors. Moreover, one could use the Bayesian θ posterior straight to infer the parameters' confidence intervals and infer possibly nonlinear correlations among individual parameters whereas MLE has to rely on variance approximations.

The disadvantage is that it usually takes much more time to fit a Bayesian model. Also, the result contains samples of the θ distribution, which may take a lot of disk space. Recent advances in the sampling algorithms, and in general having more computing power, have made it applicable to larger data sets. Currently the rule of thumb is that it is useful for data sets with at most tens of thousands of data points.

For a more detailed comparison of the Bayesian and MLE approaches please refer to the outstanding blog post "Frequentism and Bayesianism," cited above.

SIMULATION

In this article we will consider a hypothetical problem in the context of a variable annuity (VA) product and apply the Bayesian approach. Simulation and visualization are done in R, whereas Bayesian inference is done using a probabilistic language, Stan.[3] All the code, including the main Jupyter notebook, can be found on GitHub.[4] This may serve as a good-use case example for the reader.

We are going to simulate the following toy model. We will have 100 or 1,000 simulated people in our study. For each person we have 10 consecutive observations. Each observation is a binary event (i.e., 1 or 0), whether a person took a withdrawal in the given quarter or not.

The 100 or 1,000 simulated people samples will have the same random number seed, so that for the first 100 people both samples are identical. This is so that we can observe how adding more data helps in the inference of parameters.

Each person has a base withdrawal probability that can be different from other people. For example, maybe there is another parameter (income or credit score) that we do not have data for that affects the person's withdrawal probability. In our simulation, the base withdrawal probability is drawn from a normal probability density function (PDF) with a predefined mean and variance. Once drawn, it stays the same for this person. However, we allow the withdrawal probability to change with time (quarter number dependence).

To make the model more realistic, we will also allow the probability of withdrawal in a given quarter to depend on the pattern observed before that. Namely, the logit probability will have an instantaneous jump right after the first withdrawal event. This is to simulate the fact that once the first withdrawal happens, there is much higher probability that the person would withdraw in the next quarter than before that.

A one-person simulation algorithm is as follows. With predefined overall constants μ, σ, *CWD* and *Cq:*

1. Draw base logit probability from normal distribution:

$$\text{base\_logit} \sim N\left(\ , \sigma^2 \right)$$

2. Initialize withdrawal indicator $WD_{IND} = 0$
3. Loop q from 1 to 10
   - Calculate quarterly withdrawal probability (quarter count starts from 1):

$$\frac{1}{1+\exp\left(-\text{base\_logit} - C_q\left(q-1\right) - C_{WD}WD_{IND}\right)}$$

   - Draw the resulting withdrawal observation (0 or 1) for the current quarter from Bernoulli distribution:

$$WD_q \sim Bernoulli\left(p_q\right)$$

   - If WD_q == 1, then set $WD_{IND} = 1$. If the first withdrawal happens, set the indicator to 1.

Both μ and σ define the base logit withdrawal distribution, *Cq* defines withdrawal probability dependence on quarter number and *CWD* defines an instantaneous jump in the withdrawal probability after the first observed withdrawal.

We are interested in the inference of overall model constants μ, σ, *CWD* and *Cq*, as well as base logit probabilities for individual people.

EXPLORATION

In Figure 1, one can see the simulated withdrawal probabilities and withdrawal events for the first two simulated people. The first person turned out to have a much lower base withdrawal probability than the second person, by about 40 percent. We can also observe this effect in the observed withdrawal events. The second person has a higher number of withdrawals: eight versus four.

The first person's first withdrawal happens in the third quarter. For the second person, the first withdrawal is in the second quarter. Right after the first withdrawal we can observe an instantaneous jump in the simulated withdrawal probabilities for both of them, by about 6 percent in this case.

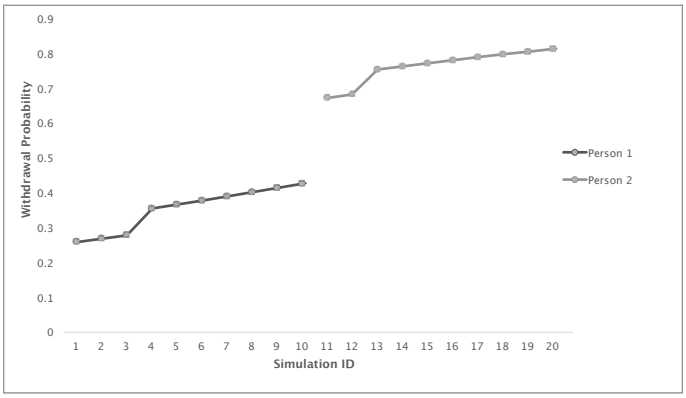
There is a roughly linear increase in probability of withdrawal with the quarter number.

BAYESIAN MODEL AND RESULTS

For the Bayesian inference we used Stan. Stan has an interface with R, the rstan package. When using rstan, one can construct data in R, launch Stan inference and get results back in R. All the code is available on GitHub. From the example, one could see that programming in Stan is fairly straightforward. All that's required is to specify the data structure, declare model parameters and specify the model—both prior and likelihood. When you pass data and the model code to Stan, it produces posterior distributions for the model parameters.

In this case we have 4+N people model parameters: four coefficients—μ, σ, *CWD* and *Cq*—and a base logit probability for each person in the training sample. Performing inference using MLE with this many parameters would be problematic because of a high chance of overfitting. However, as discussed earlier, in the Bayesian approach there is no overfitting.

Figure 1
Simulated Withdrawal Probabilities



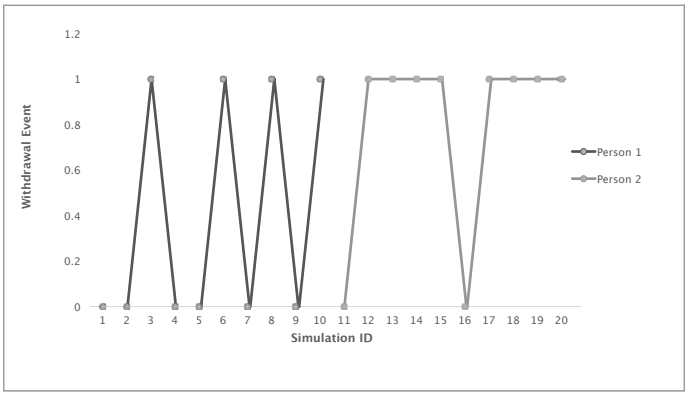


Figure 2
Inferred Distributions for μ, σ, C_{WD} and C_q

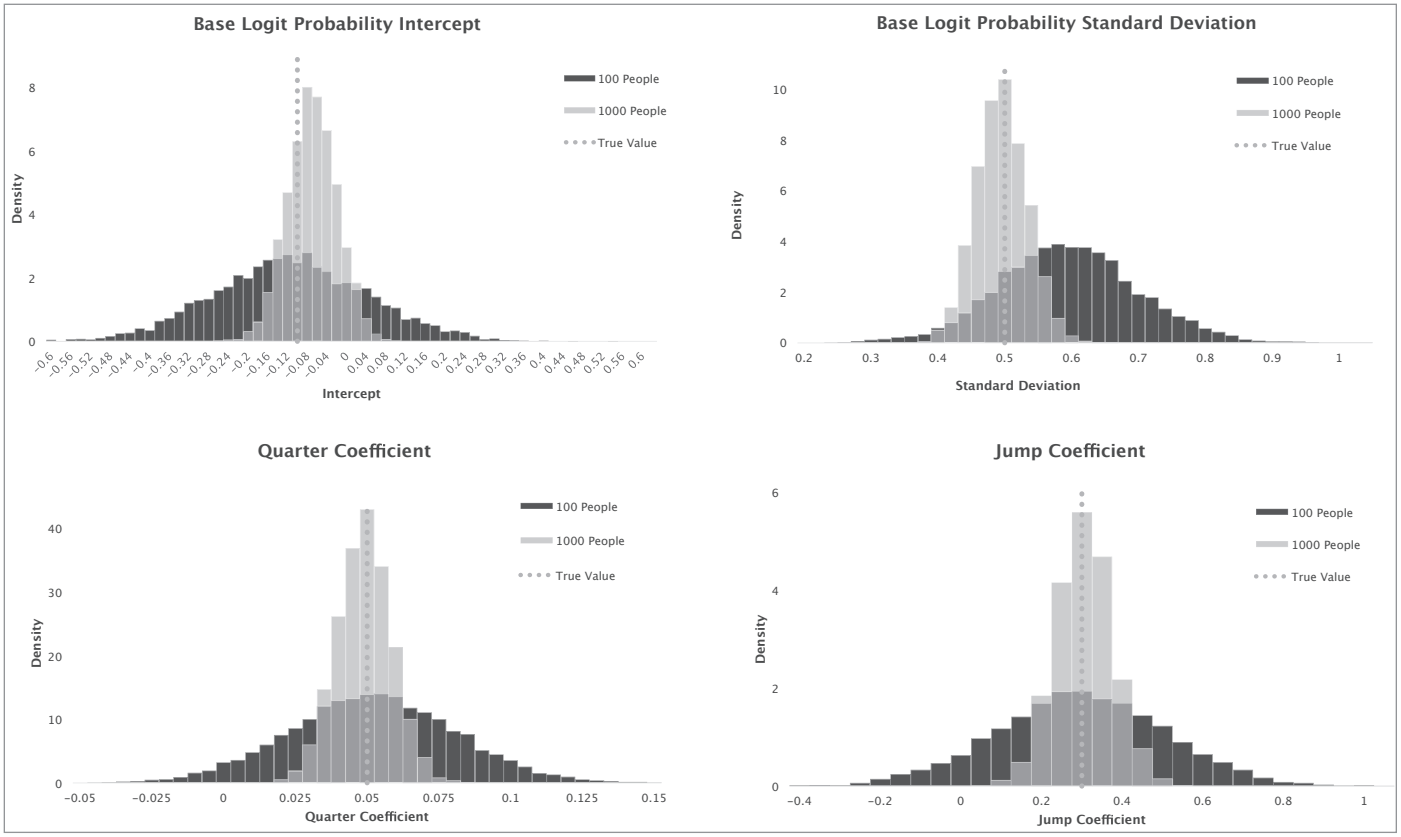
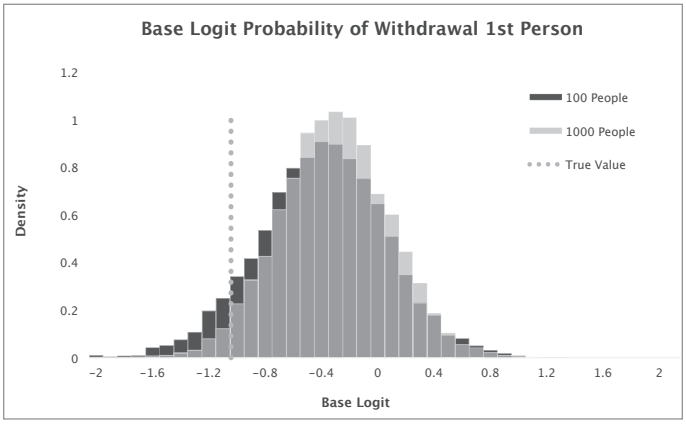
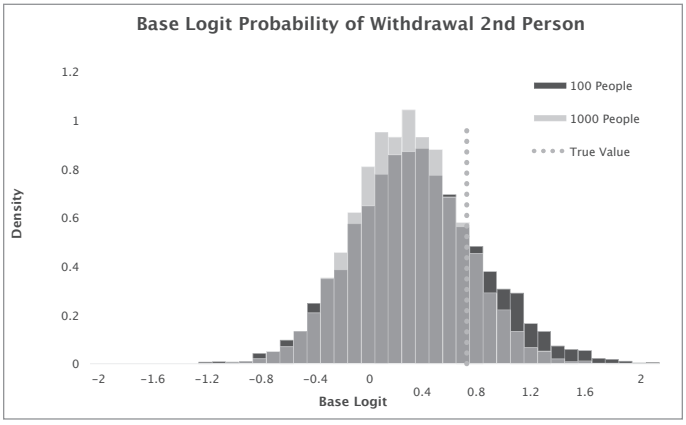


Figure 3
Inferred Distributions for Base Logit Probabilities for the First Two People in the Sample





Inferred PDFs of the four coefficients, together with their simulated true values, are shown in Figure 2. As one can see, having more data helps in the model coefficients inference—distributions become narrower.

Also, we can infer the base logit withdrawal probabilities for individual people—their withdrawal logit probabilities in the first quarter. The inferred base logit probability PDFs for the first two people in the samples are shown in Figure 3. These are the same two people from Figure 1. The two means that both sample PDFs in Figure 3 are very close, because they are the same two individuals in the two samples. These particular people have the same observations in 100- and 1,000-people samples, in terms of both total number of withdrawals and the withdrawal pattern from Figure 1. The true values in Figure 3 are a bit off from the means, because the first person had more withdrawals than expected and the second person had slightly fewer withdrawals than expected. But these true values are still within the posterior PDFs.

As one can see in Figure 3, we do infer higher values of the base withdrawal probability for the second person than for the first person, as we observed more withdrawal events for the second person. However, the distributions are fairly wide, because we have only 10 observations for these people. We can see that using 1,000-people sample makes inferred distributions a little narrower, because we have much better inferred model coefficients. However, even if we knew those coefficients exactly, the base logit distribution for individual people would still be fairly wide, because we have only 10 observations. Thus, we conclude that, for a good inference of the individual base probability, we need more longitudinal data—more quarters.

CONCLUSION

In this article we briefly described both MLE and Bayesian approaches in machine learning, looking at their advantages and disadvantages. We then proceeded with an example toy model that may be applicable for studying VA policyholder behavior. We used a simulation so that we fully understand the input data and the underlying true model.

For Bayesian inference, we used Stan probabilistic language. All inferred distributions made sense. As the amount of training data increases, the inferred distributions become narrower and closer to the true values.

This may serve as a good example for the reader in how to use Bayesian inference. ■

Denis Perevalov is a portfolio research analyst at Milliman in Chicago. He can be reached at *denis.perevalov@milliman.com*.

ENDNOTES

1 Pythonic Perambulations, "Frequentism and Bayesianism: A Practical Introduction" (March 11, 2014), *http://jakevdp.github.io/blog/2014/03/11/frequentism-and-bayesianism-a-practical-intro/*.

2 Frequentism and Bayesianism, ibid.

3 For more information, see *http://mc-stan.org/*.

4 See *https://github.com/Denisevi4/BayesianInference*.

REFERENCES

"Bayesian Inference in Machine Learning," *https://github.com/Denisevi4/BayesianInference*.

"Frequentism and Bayesianism: A Practical Introduction," *http://jakevdp.github.io/blog/2014/03/11/frequentism-and-bayesianism-a-practical-intro/*.

Stan, http://mc-stan.org/.