Bayesian Regression on segmented data using Kernel Density Estimation

Sebastian Hönel, *Morgan Ericsson, Welf Löwe, Anna Wingkvist*
Data-driven Software and Information Quality Group – Centre for Data Intensive Sciences and Applications
Linnaeus University, 351 95 Växjö, Sweden
{sebastian.honel, morgan.ericsson, welf.lowe, anna.wingkvist}@lnu.se

Extending Bayes‘ theorem

The challenge of having to deal with dependent variables in classification and regression using techniques based on Bayes' theorem is often avoided by assuming a strong independence between them, hence such techniques are said to be naïve. While analytical solutions supporting classification on arbitrary amounts of discrete and continuous random variables exist, practical solutions are scarce.

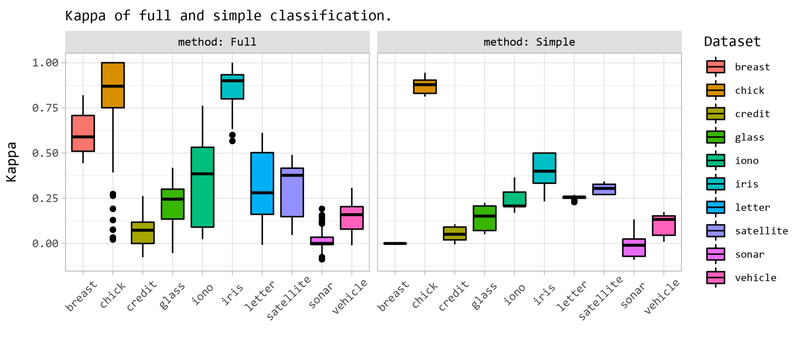
However, not only can we extend Bayes’ theorem to any number of events, we may also use continuous random variables instead, or any combination of both. Random variables however require a *conditional density*, which, in practice, is not given. We can estimate that density, by first segmenting the dataset using conditions for discrete variables, or the *empirical* conditional distribution function. This allows us to practically solve a non-naïve version of Bayes’ theorem where we assume **full** conditionality between all variables. The full general form for classification is shown below:

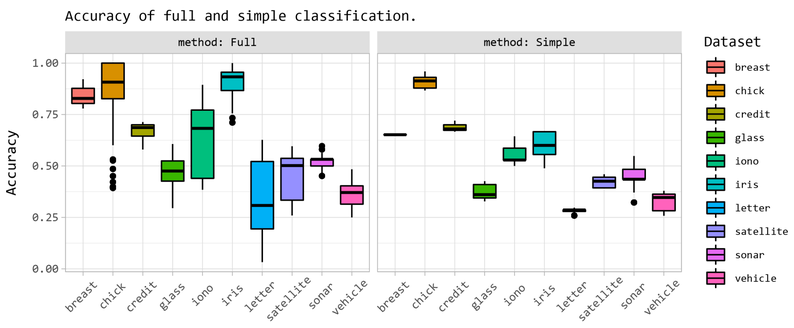
$$P(L = l|X_1 = x_1, \dots, X_n = x_n) = \frac{p(L=l)\prod_{m=1}^{n} p(X_m=x_m|X_{m+1}=x_{m+1},\dots,X_n=x_n,L=l)}{\prod_{m=1}^{n} p(X_m=x_m|X_{m+1}=x_{m+1},\dots,X_n=x_n)},$$

where each conditional $p(X_m = x_m|\dots)$ is one of

$$\begin{cases} P(X_m = x_m|X_{m+1} = x_{m+1}, \dots, X_n = x_n\ [, L = l\ ]), & \text{if } X_m \text{ is discrete,} \\ PDF_{X_m|X_{m+1}=x_{m+1},\dots,X_n=x_n\ [,\ L=l\ ]}(x_m), & \text{otherwise.} \end{cases}$$

And, accordingly, $\forall X_m \in X$ segmentation is done by $\begin{cases} X_m = x_m, & \text{if } X_m \text{ is discrete,} \\ X_m \leq x_m, & \text{otherwise.} \end{cases}$



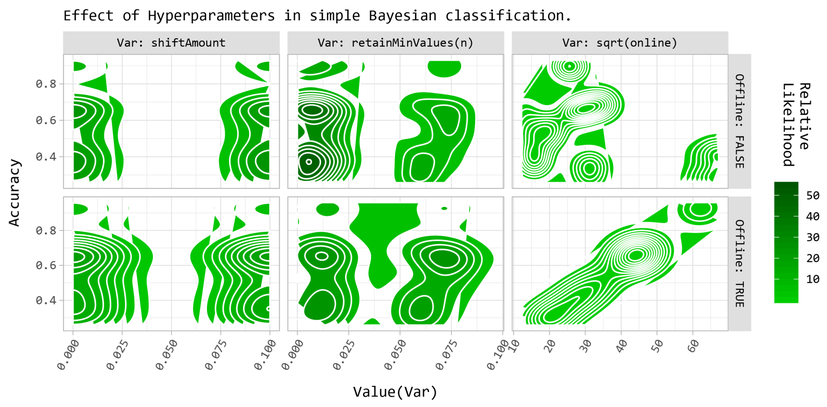


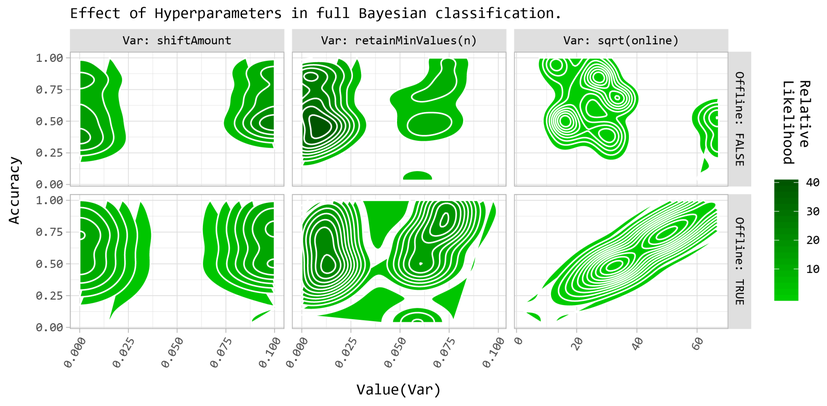
If all variables are discrete, the general form yields a probability for $L = l$, a likelihood, otherwise. This is still useful, as we can simply determine the most likely label $\hat{l} = argmax\ P(L = l|\dots) \forall\ l \in L$. This is called the *maximum a posteriori estimation* (MAPe). For each label, the strength of the class membership is also indicated. The general form can be simplified to become a Bayesian network, by waiving conditioning over the desired label. Then also, the *evidence* (denominator, normalizing constant) would become equal to the likelihood (numerator). If relinquished, the simple form looks as in the following equation:

$$Y' = P(X_1 = x_1, \dots, X_n = x_n) = \prod_{m=1}^{n} p(X_m = x_m|X_{m+1} = x_{m+1}, \dots, X_n = x_n)$$

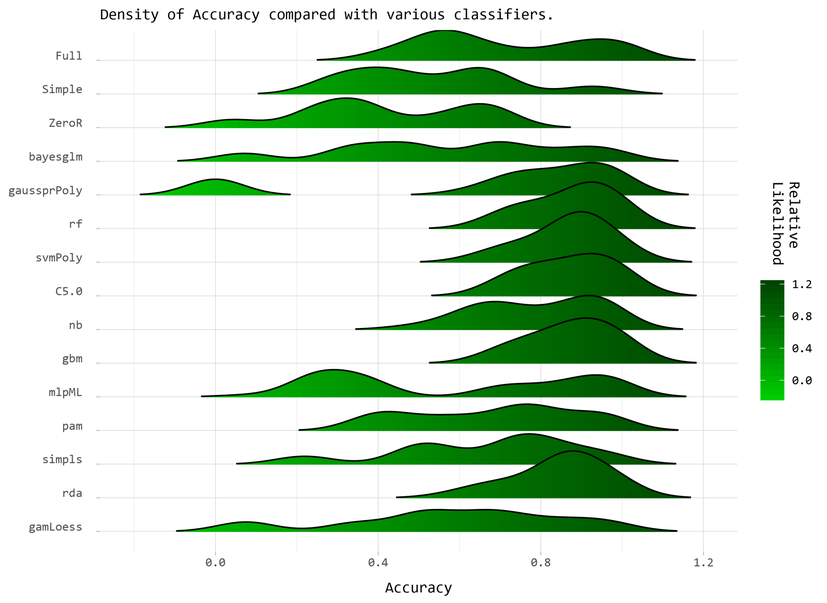
This simplified form still assumes dependence between all variables. Note that Y' holds all remaining values after segmenting. This allows us to perform a regression, by **estimating a kernel density** (KDE). Using that *empirical* PDF, similar to MAPe, a most-likely value can be obtained by finding the highest mode of the PDF, i.e., $\hat{y} = argmax\ PDF_{Y'}(y), y \in Y'$.

In order to be able to utilize the full general form, a label would be required. If our target variable were to be continuous, we can **discretize** it, and then find the most likely range. That range can be used to obtain a Y′ that also holds the most likely values for y. Using those remaining values, again, we can estimate a kernel density for **regression**. Predicting the mean would imply assuming a gaussian distribution. KDE, however, allows to estimate an arbitrary distribution from the remaining samples.
While the simplified form does not require a label, it is required for the full form. Hence, such a regression is preceded by a classification.



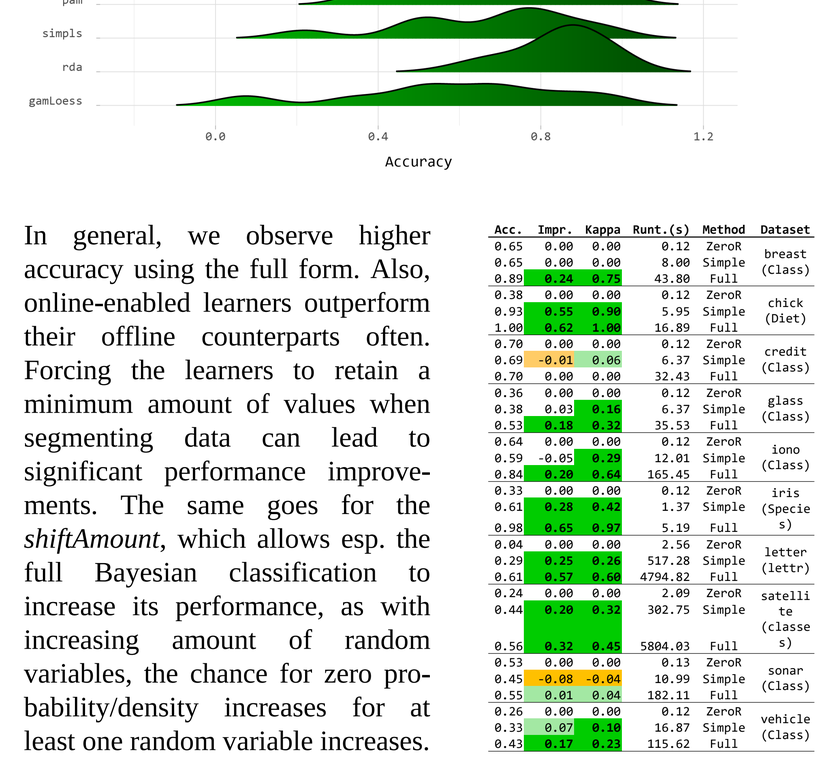


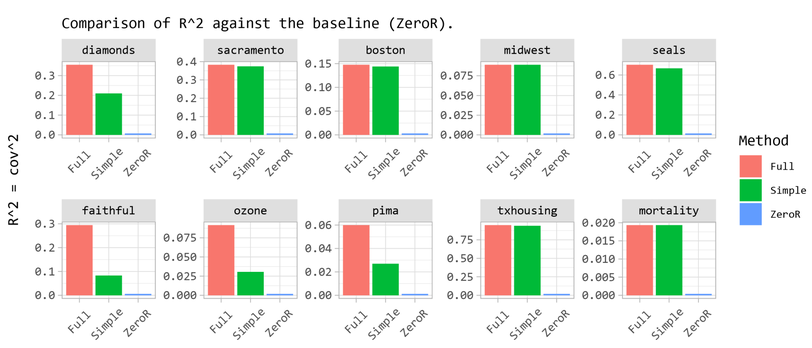
The developed models allow for a few hyperparameters. We demonstrate the effect of each w.r.t. the current performance metric, i.e., Accuracy or R^2 (the squared *Pearson* covariance).



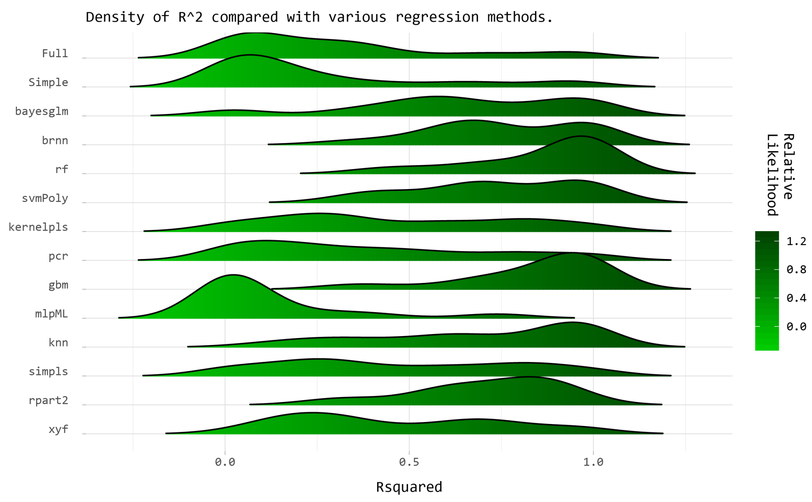
In general, we observe higher accuracy using the full form. Also, online-enabled learners outperform their offline counterparts often. Forcing the learners to retain a minimum amount of values when segmenting data can lead to significant performance improvements. The same goes for the *shiftAmount*, which allows esp. the full Bayesian classification to increase its performance, as with increasing amount of random variables, the chance for zero probability/density increases for at least one random variable increases.

Acc.	Impr.	Kappa	Runt.(s)	Method	Dataset
0.65	0.00	0.00	0.12	ZeroR	breast (Class)
0.65	0.00	0.00	8.00	Simple	
0.89	0.24	0.75	43.80	Full	
0.38	0.00	0.00	0.12	ZeroR	chick (Diet)
0.93	0.55	0.90	5.95	Simple	
1.00	0.62	1.00	16.89	Full	
0.70	0.00	0.00	0.12	ZeroR	credit (Class)
0.69	-0.01	0.06	6.37	Simple	
0.70	0.00	0.00	32.43	Full	
0.36	0.00	0.00	0.12	ZeroR	glass (Class)
0.38	0.03	0.16	6.37	Simple	
0.53	0.18	0.32	35.53	Full	
0.64	0.00	0.00	0.12	ZeroR	iono (Class)
0.59	-0.05	0.29	12.01	Simple	
0.84	0.20	0.64	165.45	Full	
0.33	0.00	0.00	0.12	ZeroR	iris (Species)
0.61	0.28	0.42	1.37	Simple	
0.98	0.65	0.97	5.19	Full	
0.04	0.00	0.00	2.56	ZeroR	letter (lettr)
0.29	0.25	0.26	517.28	Simple	
0.61	0.57	0.60	4794.82	Full	
0.24	0.00	0.00	2.09	ZeroR	satellite (classes)
0.44	0.20	0.32	302.75	Simple	
0.56	0.32	0.45	5804.03	Full	
0.53	0.00	0.00	0.13	ZeroR	sonar (Class)
0.45	-0.08	-0.04	10.99	Simple	
0.55	0.01	0.04	182.11	Full	
0.26	0.00	0.00	0.12	ZeroR	vehicle (Class)
0.33	0.07	0.10	16.87	Simple	
0.43	0.17	0.23	115.62	Full	



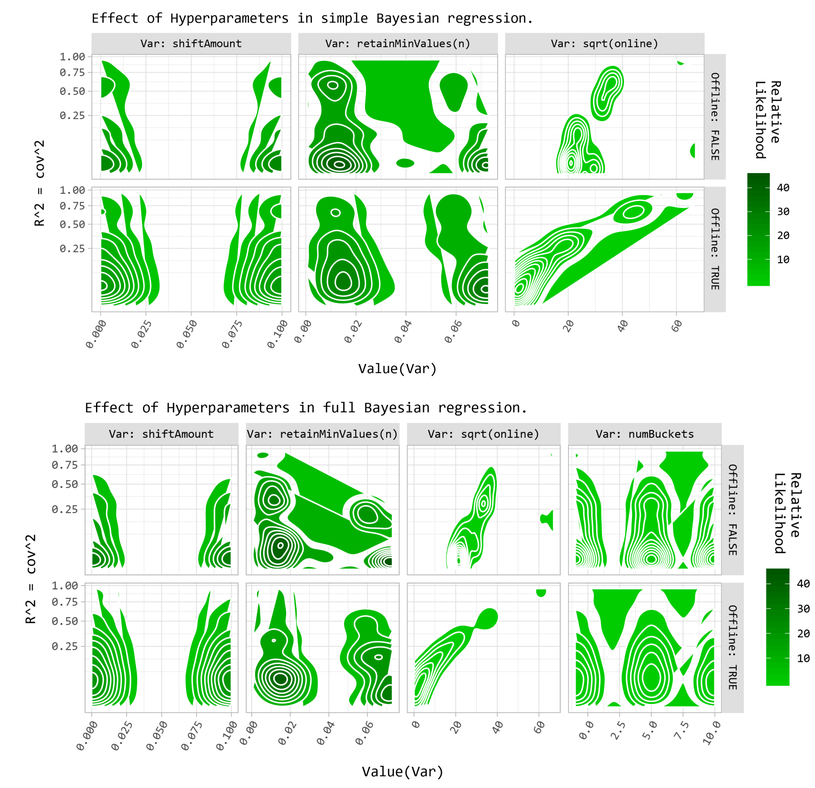


In regression tasks, the full Bayesian regression is consistently better than or equal to its simple counterpart. However, the regression models cannot currently compete with their established competitors, such as *Random Forests* or *Support Vector Machines*.



Bayesian regression using Kernel Density estimation is also met with declines in performance. Again, the full-form models outperform their simple counterparts. Retaining values leads to gains, while online-learning does not. The *shiftAmount* in the full-form models does not seem to have a significant effect, surprisingly. The number of buckets for discretization is important.

RMSE	<- Impr.	MAE	<- Impr.	Runt.(s)	Method	Dataset
2.162	0.00%	1.788	0.00%	0.11	ZeroR	boston (ptratio)
2.115	2.17%	1.523	14.81%	3.84	Simple	
2.109	2.45%	1.519	15.06%	129.04	Full	
4032.8	0.00%	3060.4	0.00%	0.12	ZeroR	diamonds (price)
4708.9	-16.77%	2873.2	6.12%	11.62	Simple	
3875.7	3.89%	2188.4	28.49%	379.25	Full	
0.008	0.00%	0.006	0.00%	0.14	ZeroR	faithful (density)
0.008	-5.22%	0.004	24.89%	4.02	Simple	
0.007	7.91%	0.004	36.44%	58.10	Full	
6.771	0.00%	4.099	0.00%	0.11	ZeroR	midwest (percwht)
6.745	0.38%	3.679	10.23%	5.56	Simple	
6.745	0.38%	3.679	10.23%	617.70	Full	
62.950	0.00%	47.936	0.00%	0.11	ZeroR	mortality (Rate)
70.987	-12.77%	42.096	12.18%	1.48	Simple	
70.987	-12.77%	42.096	12.18%	43.36	Full	
35.608	0.00%	29.636	0.00%	0.11	ZeroR	ozone (V11)
44.497	-24.96%	35.515	-19.83%	2.24	Simple	
38.034	-6.81%	30.320	-2.31%	99.55	Full	
11.751	0.00%	9.590	0.00%	0.11	ZeroR	pima (age)
15.840	-34.79%	11.054	-15.27%	4.24	Simple	
15.263	-29.88%	10.621	-10.75%	114.58	Full	
130752	0.00%	98845	0.00%	0.11	ZeroR	sacramento (price)
116770	10.69%	80989	18.06%	2.37	Simple	
110737	15.31%		23.06%	106.93	Full	
0.327	0.00%	0.270	0.00%	0.12	ZeroR	seals (delta_l)
0.205	37.39%	0.120	55.73%	1.71	Simple	
0.185	43.34%	0.101	62.78%	49.02	Full	
1134.9	0.00%	629.5	0.00%	0.14	ZeroR	txhousing (sales)
338.23	70.20%	133.7	78.76%	9.99	Simple	
294.21	74.08%	124.5	80.22%	598.36	Full	



Future Work: Neighborhood Search
Using full conditional Bayesian segmentation, we can pick any sample to segment the remaining data, which then would become its *neighborhood*. Then, for all samples, we can measure the *vicinity* to that neighborhood. This method may be used, e.g., to assess neighborhood preservation in *dimensionality reduction* techniques.

$$Vicinity_{N_i}(s'_j) = \prod_{m=1}^{n} \begin{cases} P(X'_n(j) = x_n(j)), & \text{if } X_n \text{ is discrete,} \\ PDF_{X'_n}(x_n(j)), & \text{otherwise.} \end{cases}$$

Future Work: Numerical approximation of similar and most likely conditional values (also: pseudo-random value generator).
Having a segmented dataset, a multivariate Kernel can be estimated on the remaining values for each random variable. Then, using numerical approximation, the top most similar and likely tuples can be estimated. This may be useful for, e.g., hyperparameter search or multi-goal optimization.