informs
http://pubsonline.informs.org/journal/mksc

MARKETING SCIENCE
Vol. 00, No. 0, Xxxxx 0000, pp. 000–000
ISSN 0732-2399, EISSN 1526-548X

Submitted to *Marketing Science*

Applications with Limited but Diverse Data: Improving Prediction and Uncertainty Estimation with Bayesian Deep Learning

Remi Daviet
Wisconsin School of Business, University of Wisconsin–Madison, daviet@wisc.edu,

Abstract. With the continuous increase in data availability and diversity, deep learning methods have gained in popularity, thanks to their ability to capture patterns when the type of relation between variables is unclear. An important issue with traditional deep learning methods is the need for a large number of observations for each segment of the population studied. However, in many applications, data availability is limited, either as a whole (e.g., a limited number of SKUs or stores), or in specific segments of interests (e.g., top 10% of customers). When data are limited, deep learning models tend to show poor predictive performance (overfitting), but also wrongly assess the uncertainty associated with inferences (overconfidence). In this paper, we show that adopting Bayesian neural networks (BNN) reduces overfitting, increases accuracy for categories with limited data, and produces more reliable estimations of uncertainty. Moreover, BNN facilitate tasks where the full posterior predictive is required. We resolve the computational issue of sampling from the posterior of neural networks by using a Hamiltonian Sequential Monte-Carlo approach. We demonstrate the performance of our approach with a simulation and a simple empirical example.

Key words: Deep Learning; Bayesian Inference; Uncertainty; Overfitting; Small Sample; Imbalanced Sample

Word count: 5648

1. Introduction

The last few decades have witnessed an exponential increase in the accumulation of data, which has allowed firms and practitioners to leverage the information they contain and improve forecasting, targeting, and their understanding of consumers. However, a large share of the available data is high dimensional and mixing a broad range of variables types, leading to situations where there is no clear and natural model explaining the data structure. In these cases, the data are often referred to as being unstructured. Consequently, traditional modeling approaches are difficult to use with these datasets. As an alternative, deep learning models have shown a remarkable ability to capture complex patterns, even which highly unstructured data such as product pictures, user reviews, or social network graphs. However, effectively training these models requires a substantial amount of data in every regions of interest in the data distribution. In a managerial setting, data is not always available in sufficient quantities: for many applications, expending the number of observations can be extremely costly or nearly impossible, especially for under-represented classes. For instance, a company might have a limited number of products, number of retail locations, or number of consumers qualifying as "top 1% spender." Moreover, in some cases such as with consumer segments, it might not be clear whether and which classes are being underrepresented.

One of the main issues when training flexible models with limited data is that the predictive performance in the training sample is inflated and does not generalize well to other samples, a phenomenon broadly known as overfitting (Ying 2019, Salman and Liu 2019). This is a direct consequence of using an optimization-based approach such as likelihood maximization or loss function minimization (e.g., minimizing the root mean squared error). A popular method to reduce overfitting is the use of regularized optimization (Kearns et al. 1997, Dietterich 1995), where a prior is introduced to act as a penalty term and to reduce the model's flexibility. The strength of the regularization is usually tuned to maximize out-of-sample prediction performance. While this indeed increases the model fit in new samples, we confirm in this paper that like other overfitting avoidance strategies, it can introduce bias (Schaffer 1993) and give a false sense of confidence concerning the prediction accuracy by inferring unreliable prediction errors distributions (Wang et al. 2021, Jiang et al. 2018).

Another issue is that different parametrizations of a model often fit nearly equally well observations in the dense part of the data distribution, but might widely differ for predictions in regions with few or no observations, a phenomenon known as underspecification (D'Amour et al. 2022). The use of an optimization approach with regularization leads to the selection of a specific parametrization with acceptable predictions for typical observations, but with potentially poor accuracy in regions of the data distribution that are underrepresented. This is a major issue as managers might specifically be interested in these rarer cases. For example, typical marketing tasks may include accurately predicting which products would be in the top 5% of performers, exploring unusual combinations of attributes for new product offerings, or understanding why the bottom 5% of retail locations are under-performing. A potential approach, known

as data augmentation, attempts to reduce overfitting while avoiding the issues associated with strong regularization by artificially creating new observations, adding noise or modifications to existing data points (e.g., mirroring a product picture) (Karras et al. 2020). Data augmentation is however not always feasible and might actually increase overfitting: for instance, adding several stores to a dataset with slightly different characteristics but similar outcomes could lead the model to be biased toward these stores' outcomes.

Beyond potential inaccuracies and bias, another issue with deep learning is that most models only provide a prediction with no clear strategy to estimate the associated uncertainty, which is critical in many managerial applications. Knowing whether a sales forecast is subject to a $\pm 5\%$ or a $\pm 20\%$ margin of error has important implications. Businesses could favor pursuing an option with low uncertainty to another with higher expected gain but more risk. Uncertainty estimation can be addressed with the use of probabilistic models, where the neural network provides a full predictive distribution instead of a simple point prediction. In the case of limited data availability, this distribution might be inaccurate, as shown in some of this paper's examples. Specifically, with low regularization, the model strongly overfits and provides an underestimation of uncertainty regarding potential outcomes, a phenomenon known as overconfidence. An extreme case of overfitting and overconfidence is to predict all the in-sample observations with near-perfect accuracy and infer zero variability in the outcomes (perfect confidence). With stronger regularization, the model might on the contrary predict a larger outcome variability than necessary.

In the case of modern neural networks, systematic overconfidence and miscalibrated uncertainty estimations have been documented, especially with cases underrepresented in the training data (Hein et al. 2019, Guo et al. 2017). Moreover, regularized optimization approaches attribute all the uncertainty to outcome variability, traditionally represented by an error term's variance. This approach fails to consider another source of uncertainty which is associated with the model parameters themselves, especially with complex and flexible deep learning models. For a large share of published research, an emphasis is put on increasing accuracy without distinguishing between these sources of uncertainty, and without explicit attempts to measure them separately. Managers might however be interested in distinguishing the uncertainty coming from outcome variability, which is due to unobservable factors and cannot be avoided, from model uncertainty, which can be mitigated with additional data.

We propose to solve these issues by adopting a Bayesian perspective and developing a method sampling from the full posterior distribution of the neural network parameters instead of using regularized maximization approaches. While theoretically Bayesian neural networks appears to be good candidates to solve the issues mentioned above, their ability to tackle them in practice remains unclear. We show in this paper that using Bayesian neural networks substantially mitigates the overfitting issue and reduces bias, especially for small samples and underrepresented cases. We also show that this approach provides a more accurate estimation of model uncertainty and outcome variability. Fitting a Bayesian deep learning model to a dataset is however not a simple task: several computational issues arise from the high-dimensionality and heavy

multimodality of the posterior distribution (Li et al. 2018). To efficiently sample from this complex posterior distribution, we adopts a Hamiltonian Sequential Monte-Carlo (HSMC) approach, which combines the advantages of Hamiltonian Monte-Carlo's gradient-informed sampling (Neal 2012) with sequential prior updating (Del Moral et al. 2006). The combination of Hamiltonian Monte-Carlo and Sequential Monte Carlo technologies, have shown promising results for lower dimensional models with complex likelihoods (Burda and Daviet 2022, Buchholz et al. 2021). Another advantage of this approach is the ability to leverage automatic differentiation technology largely used in deep learning (Baydin et al. 2017). To our knowledge, it is the first paper where Hamiltonian Sequential Monte-Carlo methods are adapted to fit deep learning models. Furtermore, we are not aware of other papers using Bayesian neural networks in Marketing.

We compare our proposed approach with traditional regularized optimization where the strength of the regularization is tuned via cross-validation to maximize the out-of-sample likelihood (Cooil et al. 1987). Cross-validation artificially creates different samples using resampling or partitioning methods, allowing practitioner to use the different parameter estimates across samples as a proxy for model uncertainty. A disadvantage of cross-validation methods is that, unlike our proposed method, the required data partitioning further reduces the amount of observations available for training. When the dataset is small, this additional reduction in training sample size might amplify issues associated with regularized optimization. We also test a hybrid training procedure where we adopt a Bayesian approach, but tune the concentration of the prior distribution to maximize the out-of-sample likelihood using cross-validation fashion, which also requires to split our sample into training and validation sets.

We first perform a simulation study where the ground truth is known, and assess the performance of each method for various sample sizes. We find that compared to our method, both of the cross-validation approaches do not estimate uncertainty accurately. In particular, they overestimate outcome variability (variance of the error term) and underestimate model uncertainty, with the true value of the outcome variability far out of the confidence interval. By contrast, our approach produces a substantially more accurate estimate of outcome variability, with the true value falling within confidence bounds. To compare the accuracy of each model for marginal cases, we compute the proportions of observations in the test sample that have been correctly predicted to belong to the top 10%. Our approach drastically outperforms its competitors, achieving a 40% accuracy with samples as small as $N = 3,000$, where other approaches remain below 20% even in samples of $10,000$ observations. We then perform a similar exercise with a public dataset of vacation rentals, where the ground truth is unknown, and confirm that the Bayesian approach provides better RMSE for smaller sample sizes.

We believe that our paper presents a substantial contribution to the many applications where data are high-dimensional or without clear structure but observations are limited, ultimately allowing practitioners and researchers to leverage new sources of data with deep learning methods that remained inapplicable until now. The ability of our method to provide a better estimate of uncertainty also allows practitioners to make decision with knowledge of the risk involved.

2. General Model and Method

In this section we define the general Bayesian model and associated estimation problems. We describe three estimation methods: a traditional optimization-based approach with cross-validation, a fully Bayesian approach, and a hybrid Bayesian approach where the prior is tuned to maximize out-of-sample performance.

2.1. Model

We denote the vector of explanatory variables as x with distribution $p(x)$, and we are interested in the conditional distribution of the dependent variable $y \sim p(y|x)$. While most applications put a strong emphasis on recovering the first moment of the conditional distribution $\mu_{y|x} = E[y|x]$, we also want to have an estimate of the uncertainty associated with outcome variability, often quantified through a conditional scale parameter or standard deviation $\sigma_{y|x}$. We thus define a probabilistic parametric model $(\mu_{y|x}, \sigma_{y|x}) = g(x;\theta)$ where the function $g(\cdot)$ maps x to a value of $(\mu_{y|x}, \sigma_{y|x})$ given the parameter θ. In a deep learning setting, this mapping typically belongs to a class of neural networks such as a series of fully connected layers (multilayer perceptron), a convolutional neural network, or a long short-term memory model (LSTM). In these cases, each layer in the neural network has its own set of parameters $\theta = (\theta_1, ..., \theta_K)$, where K is the number of layers. We adopt a Bayesian framework where the uncertainty about the parameter θ_k is represented by a prior distribution $p(\theta_k)$ such that $\theta_k \sim \mathcal{N}(0, (|\theta_k|_{in} \cdot \lambda_k)^{-2})$, where λ_k is the regularization parameter and $|\theta_k|_{in}$ is the number of inputs for layer K. This distribution corresponds to a RIDGE regularization penalty and the standard deviation being inversely proportional to the number of inputs corresponds to the He initialization of neural networks (He et al. 2015).

2.2. Estimation Methods

In this section we first by discuss the traditional machine learning approach of training through dataset splitting and regularized optimization. We also describe how parameter uncertainty can be estimated with cross-validation. We then proceed to present our Bayesian approach and its advantages.

Traditionally the dataset $\mathcal{D} = \{y_i, x_i; i = 1, ..., J\}$ is divided into three subsets: a *training set* $\mathcal{D}_{train}$, used to fit the model; a *validation set* $\mathcal{D}_{val}$, used to adjust the regularization strength; and a *test set* $\mathcal{D}_{test}$, used to measure out-of-sample performance. Specifically, the model is fitted to the training set $\mathcal{D}_{train}$ using a stochastic gradient descent algorithm optimizing an objective function. Since we are interested in providing both point predictions and associated outcome uncertainty, minimizing the classic functions of prediction errors (Sum of Squares, etc.), is not sufficient here as it only takes the point prediction into account. We instead maximize the likelihood, regularized by a prior, as this function is both penalized by large errors and by wrong variance estimates. In effect, we obtain a *maximum a posteriori* (MAP) parameter estimate $\hat{\theta}$ such that:

$$\hat{\theta} = \arg\max_{\theta} p(y_{train}|\theta, x_{train})p(\theta) \tag{1}$$

$$= \arg\max_{\theta} \log p(y_{train}|\theta, x_{train}) + \log p(\theta), \tag{2}$$

where $\{y_{train}, x_{train}\}$ are the observations in $\mathcal{D}_{train}$. With that approach, the vector of regularization parameters λ is set to maximize the predictive performance in the validation set $\mathcal{D}_{val}$, which can be measured with metrics such as the coefficient of determination R^2 or the validation-sample likelihood $p(y_{val}|\hat{\theta}, x_{val})$. Finally, the overall out-of-sample performance after training is measured in the set $\mathcal{D}_{test}$, which was neither used for training nor for parameter tuning. As most objective functions for deep learning models are highly irregular, with many local optima (Li et al. 2018), selecting only the best fitting parameter value might ignore several other highly likely values that would perform well out-of-sample. Moreover, optimization is usually done through gradient descent methods and have a high risk of converging to a local optimum. These issues get magnified in small samples, creating a significant risk of overfitting unless strong regularization is used, which impairs performance and biases estimates (Marquardt 1980).

While this regularized optimization approach provides us with an estimate of outcome variability through $\sigma_{y|x}$, it does not take into account the uncertainty relative to the accuracy of the parameter estimate $\hat{\theta}$. This uncertainty can be assessed using cross-validation methods, where the observations are randomly shuffled across $\mathcal{D}_{train}$ and $\mathcal{D}_{val}$, providing us with many partitions $\{\mathcal{D}_{train}^{(j)}, \mathcal{D}_{val}^{(j)}\},\ j = 1, ..., J$, of the dataset. The model is then repeatedly fitted to each of these partitions, and the set of corresponding MAPs $\hat{\theta}^{(j)},\ j = 1, ..., J$, can be used to represent the parameter uncertainty.

In our Bayesian framework, we do not need to tune the regularization parameter and thus split the dataset in only two subsets: a *training set* with the "in-sample" observations $\mathcal{D}_{in}$ used to fit the model; and the *test set* $\mathcal{D}_{test}$ which is identical to the regularized optimization case. The uncertainty relative to the value of θ is directly represented in the posterior distribution:

$$p(\theta|y_{in}, x_{in}) = \frac{p(y_{in}|\theta, x_{in})p(\theta)}{\int p(y_{in}|\theta, x_{in})p(\theta)d\theta}, \tag{3}$$

where $\{y_{in}, x_{in}\}$ are the observations in $\mathcal{D}_{in}$. The posterior distribution being complex and often difficult to recover, we propose in the next section a novel Monte-Carlo method to sample from it.

Finally, we also evaluate a hybrid approach where the model is fitted to $\mathcal{D}_{train}$ following a Bayesian approach as described above, but where the regularization parameter λ is tuned to maximize the predictive performance in the validation set $\mathcal{D}_{val}$. Since this regularized Bayesian approach provides samples from the full posterior, no cross-validation is necessary to provide an estimate of model-uncertainty.

Each of the three methods described above is evaluated on the same test set $\mathcal{D}_{test}$ which allows us to directly compare their performance and limitations.

2.3. Bayesian Posterior Sampling

The simulation of the posterior distribution presents several computational challenges. The number of parameters to estimate is potentially high and the likelihood complex, making it a difficult high-dimensional simulation problem. Hamiltonian Monte-Carlo (HMC) methods have been shown to be particularly appropriate for Bayesian Neural Network models (Neal 1996). This method uses the gradient of the posterior

distribution to improve the efficiency of the random sampling. These gradients are easily obtained with most deep learning software packages relying on the autograd approach for differentiation (Baydin et al. 2017). Unfortunately, HMC is not very effective with multimodal target distributions and we thus propose to use the Sequential Hamiltonian Monte-Carlo (SHMC) algorithm (Burda and Daviet 2022), where several HMC chains are run in parallel. This method allows us to sample around multiple modes, as various chains can converge to different regions. In addition, HMC allows the observations to be included sequentially in the algorithm, with the model being fitted to a progressively increasing number of observations, corresponding to a sequence of (usually progressively sharper) posterior distributions. The progressive injection of data in the training set, sometimes referred to as data tempering, saves computational time by only processing a few data points at first, when only a rough estimate of the posterior distribution is needed, before including more and more data points as the sampling distribution converges to the true posterior. To our knowledge, this paper is the first to demonstrate the usability of HSMC methods in a deep learning context.

2.4. Simple Illustrative Example

We provide here an extremely simple and intuitive example to illustrate how the Bayesian approach can be advantageous when performing inference with small samples. We generate a very simple dataset of 30 observations with one explanatory variable $x_i \sim \mathcal{N}(2.5, 1)$ and one dependent variable $y_i \sim \mathcal{N}(x_i, 0.5^2)$, such that $E[y_i|x_i] = x_i$. We then fit a standard linear model $y_i = \beta_0 + \beta_1 x_i + e_i$ with both the cross-validated MAP approach and the Bayesian approach using a flat prior (unregularized). Cross-validation is performed over $K = 20$ folds in the MAP case, providing us with 20 separate estimates. For comparison purposes, we also sample 20 estimates from the posterior distribution in the Bayesian case.

We provide in Figure 1 a representation of the regression lines for both approaches, as well as the distribution of the estimated β_1. For visualization purposes, we also provide a representation of the pseudo-likelihood $p(y|\beta_1)$, computed as $\exp(-\frac{1}{K}\sum_i(\frac{y_i-\beta_1 x_i}{0.5})^2)$. We can see that all the estimates in the MAP case are close to the "most likely" value for β_1, whereas the Bayesian estimates are more spread out, following the posterior distribution. This increased diversity obtained from the Bayesian approach provides a better representation of the model uncertainty. We particularly note that while the regression lines for both methods predict roughly the same values in the range $x \in [2, 3]$ where a lot of observations are available, they differ substantially in the tail of the distribution for extreme values of x, where uncertainty is greater. This illustrates how a Bayesian approach can better capture model uncertainty, specifically in the tails of the data distribution.

3. Simulation

In this section, we simulate a simple dataset where the ground truth is known. We then fit a neural network using each of the approaches described in Section 2.2 with training sample sizes ranging from 100 to $10,000$ observations. The performances in terms of prediction accuracy are finally compared on a identical test set

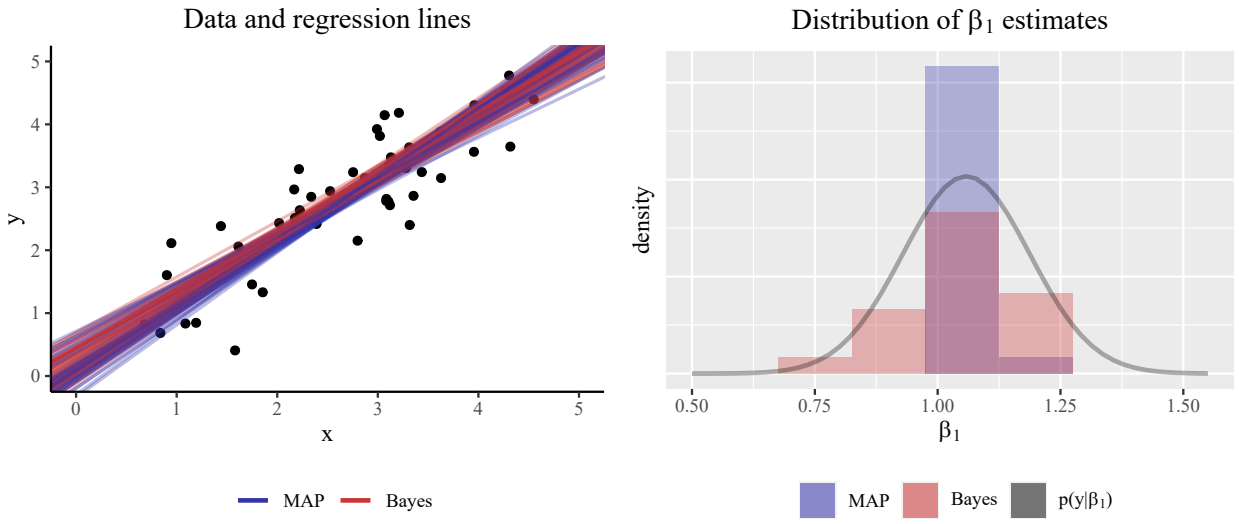

Figure 1 **Left: data points and regression lines from cross-validated MAP and Bayesian posterior draws. Right: distribution of β_1 estimates from the cross-validated MAP and the Bayesian posterior, as well as a representation of the pseudo-likelihood $p(y|\beta_1)$.**

of $2,000$ observations. The simulation code is available at `https://osf.io/ju56w/?view_only=4c9143bb741f4328a159530200b1305f`.

3.1. Data and ground truth

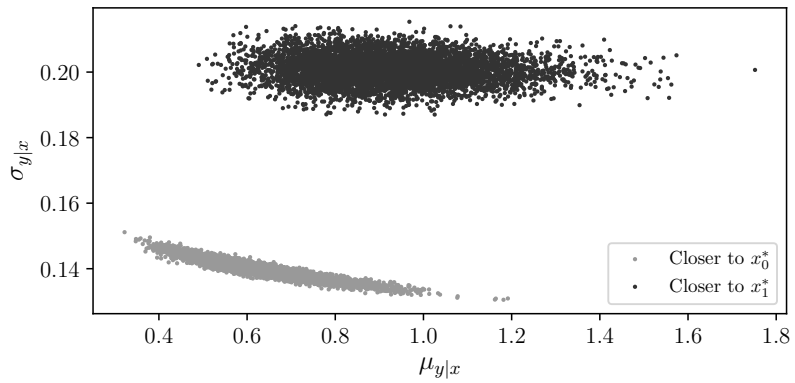

Figure 2 **Joint distribution of $(\mu_{y|x}, \sigma_{y|x})$ in our simulated dataset.**

We construct a dataset with a total of $12,000$ observations, where 100 to $10,000$ observations are progressively used for training, in increments of 100 observations. These observations are either allocated to $\mathcal{D}_{in}$ in the Bayesian case, or to $\{\mathcal{D}_{train}, \mathcal{D}_{val}\}$ with an 80%-20% split in cases requiring a validation set for tuning. The other $2,000$ observations are used in $\mathcal{D}_{test}$ for out-of-sample performance evaluation. Once the model is trained, the out-of-sample performance is tested using $\mathcal{D}_{test}$.

The explanatory variables x have a dimension of 128 and are drawn from either $\mathcal{N}(0, I \cdot 0.5^2)$ or from $\mathcal{N}(1, I \cdot 0.5^2)$ with equal probabilities, generating a bimodal distribution with modes at $x_0^* = [0, ..., 0]$ and $x_1^* = [1, ..., 1]$. The dependent variable y is set to have a high expected value at each of these modes and to exponentially decrease as the distance to the mode increases. This multimodality could correspond to a market where there are two families of products (e.g., budget and premium) with characteristics x potentially targeting different segments of consumers and associated sales y. As the products are designed to fit consumers' preferences, more products are offered in regions where expected sales are higher.

We set the y at the second mode to have a higher expected value, but also shows a lager outcome variability. More formally, the expected value and the standard deviation for y are calculated as follow:

$$\mu_{y|x} = 10 * e^{-0.5 \cdot ||x - x_0^*||_2} + 15 * e^{-0.5 \cdot ||x - x_1^*||_2}, \tag{4}$$

$$\sigma_{y|x} = 0.1 + 0.1 \cdot \frac{1}{128} \cdot ||x - x_0^*||_2, \tag{5}$$

where $|| \cdot ||_2$ is the Euclidian norm. The resulting distribution of $(\mu_{y|x}, \sigma_{y|x})$ is shown in Figure 2. A value for y is then randomly drawn from $\mathcal{N}(\mu_{y|x}, \sigma_{y|x})$ for each x in the sample.

3.2. Neural Network Model

The model used is a neural network with 5 fully connected layers of respective dimensions $[128, 16, 8, 4, 2]$. We chose the neural network architecture that gave the best RMSE in the validation sample for the MAP approach to avoid giving an unfair advantage to our method. Increasing the depth (number of layers) or the width (number of nodes in each layer) did not improve the predictive performance of the neural network in our tests. On the contrary, a more flexible neural network becomes harder to fit and increases the issues met with small datasets. On the other hand, smaller neural networks resulted to decreased predictive performance due to a lack of flexibility. The activation function for intermediate layers is a leaky ReLU function with a slope of 0.1 for negative inputs, and 1 for positive inputs. The final layer presents 2 outputs, one not subject to any activation function, allowing the predicted μ to take any value in $\mathbb{R}$, and one subject to an exponential activation function to ensure that σ remains positive.

3.3. Simulation results

We train our model using each of the method described in Section 2. For the regularized optimization case, we tune the regularization parameter to maximize the validation set likelihood, averaged across 128 cross-validation partitions.

We report in Figure 3 two measures of predictive performance for the three tested approaches. The first measure is the average out-of-sample likelihood, which provides an estimate of the overall fit of the model. For the optimized MAP case, the average is taken over the cross-validation samples. For the other cases, it is taken over Monte Carlo draws from the posterior, respectively representing the marginal predictive likelihoods $p(y_{test}|X_{test}, \mathcal{D}_{in})$ for the Bayesian approach and $p(y_{test}|X_{test}, \mathcal{D}_{train})$ for the hybrid approach.

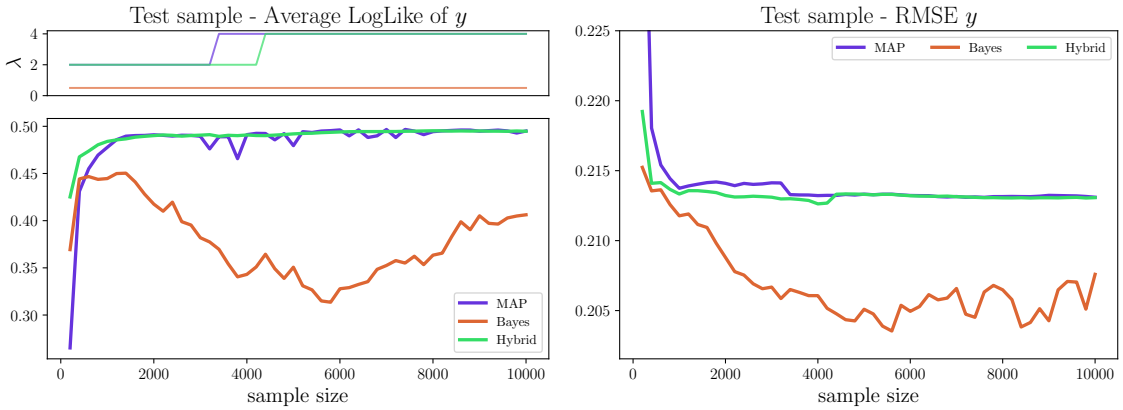

Figure 3 **Various measures of out-of-sample predictive performance for simulated samples ranging from 100 to 10,000 observations.**

As expected, the optimization and hybrid approaches have a better overall out-of-sample likelihood, as their regularization parameter λ have been tuned to maximize this specific measure of performance. By contrast, our Bayesian approach performs better when looking root mean square error (RMSE), with a RMSE revolving around 0.205, whereas the other methods remain near 0.215. For reference, the average $\sigma_{y|x}$ in our sample is at 0.175 which would correspond to a theoretical minimal RMSE.

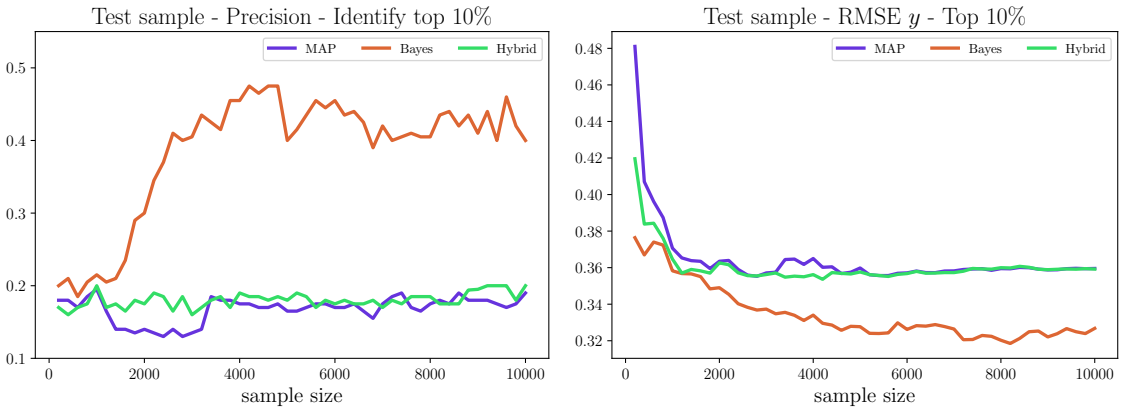

Figure 4 **Various measures of out-of-sample predictive performance for the top decile of y values in simulated samples ranging from 100 to 10,000 observations.**

We then provide two measures predictive performance for extreme cases, represented by the top decile of y values in the test sample (Figure 4). The first measure is classification precision, where we compute the proportion of observations correctly predicted to be in y's top decile. An observation is predicted to be in the top decile if the average predicted $\mu_{y|x}$ across Monte-Carlo samples / cross-validation estimates is in the top decile of predictions. The Bayesian approach drastically outperforms its competitor, achieving a 40% accuracy with samples as small as $N = 3,000$, where other approaches remain below 20% even

with samples of $10,000$ observations. This illustrates the fact that even with larger samples, extreme cases remain poorly predicted with regularized methods. A second measure of accuracy is the RMSE for these extreme y values, where our Bayesian method also shows an advantage. In an example where x represents product characteristics and y represent sales, correctly identifying which products would perform in the top sales decile would give a business a certain competitive advantage.

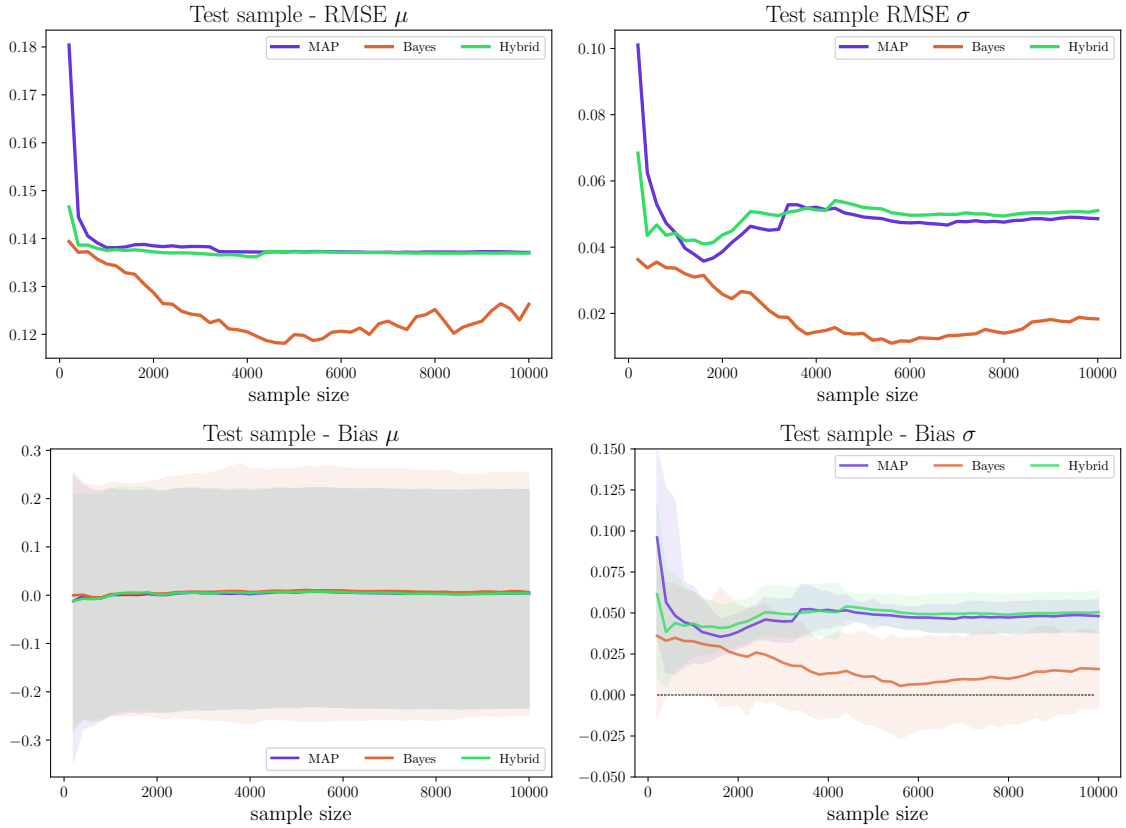

Figure 5 **Out-of-sample estimation accuracy (top) and bias (bottom) in $\mu_{y|x}$ (left) and $\sigma_{y|x}$ (right) using sample sizes ranging from 100 to 10,000 observations. The shaded area represents the estimated uncertainty at a 95% confidence level.**

We then assess the model ability to estimate the parameters $(\mu_{y|x}, \sigma_{y|x})$ of the distribution of y for each observation in the test set (Figure 5). The first measure considered is the RMSE of the predicted parameter values against the true parameter values. Our Bayesian approach rapidly outperforms other approaches both in estimating the mean $\mu_{y|x}$ and the outcome variability $\sigma_{y|x}$. We then consider the average bias over observations, which is simply the difference between the parameter predicted value and the true value. The bias is computed for each Monte-Carlo sample in the Bayesian cases, and for each cross-validation estimate in the optimization case. Regarding the mean parameter $\mu_{y|x}$, all 3 approaches appear on average unbiased. However, our Bayesian approach has a lower bias when estimating $\sigma_{y|x}$, with the ground truth being withing

confidence bounds. The regularized approaches, on the other hand, show a higher bias for $\sigma_{y|x}$ and do not cover the ground truth within their confidence bound. This confirms our Bayesian approach outperforms the regularized approaches in estimating the uncertainty associated with a prediction.

With this simulation, we have shown how a Bayesian neural network can predict outcomes with better accuracy in small samples, is substantially better at identifying extreme cases, and provides better estimates of the outcome distribution characteristics, including outcome variability. All these characteristics can be useful for businesses across many tasks, from forecasting to segmentation.

4. Applied Example: Vacation Rental Market Price

We provide here a simple example based on a public dataset of vacation rentals for the city of Toronto in September 2022. The data is provided by Inside Airbnb, a project with the objective to quantify the impact of short-term rentals on housing and residential communities. We want to highlight that the goal of this section is only to compare the performance of each method with a simple example, not to perform state-of-the-art analysis of the dataset. We are aware for instance that other control variables could be added, and that causal or time series inference methods could be used. We voluntarily keep our example as simple as possible and leave more advanced analyses to other papers dedicated to this specific topic.

Category	Variables
Host characteristics	Days active, Superhost, Verified, Number of listings
Property characteristics	Capacity, Entire/shared home, Property type (Condo, Rental, Home, Suite, Townhouse), # beds, # bedrooms, # bathrooms, Shared bathroom, Latitude, Longitude.
Amenities	Kitchen, Wifi, Essentials, Hair dryer, Heating, AC, Hot water, Shampoo, Conditioner, Dishes/silverware, Dishwasher, Refrigerator, Cooking basics, Washer, Dryer, Iron, Microwave, Stove, Oven, Freezer, Table, TV, Dedicated workspace, Parking on premises, Street parking, Paid parking, Bathtub, Gym, Patio/balcony, BBQ/grill, Backyard, Pool, Hot tub, Lake access
Other	Min nights, Total reviews, # reviews (30 days), Review score

Table 1 Explanatory variables

Our dataset has 55 explanatory variables (see Table 1), which we use to predict the log market price. Market price prediction can be useful for the rental platform to give pricing recommendations to hosts. All the variables are standardized prior to training. The dataset has about 10,767 observations, and we use 8,000 observations for training, by increments of 250, and 2,000 for out-of-sample testing. The data are preprocessed following a method described by Lewis (2019). We then train a neural network similar to the simulation part but with a $[55, 32, 16, 8, 4, 2]$ architecture, as it provided the best RMSE in the validation set using the MAP method.

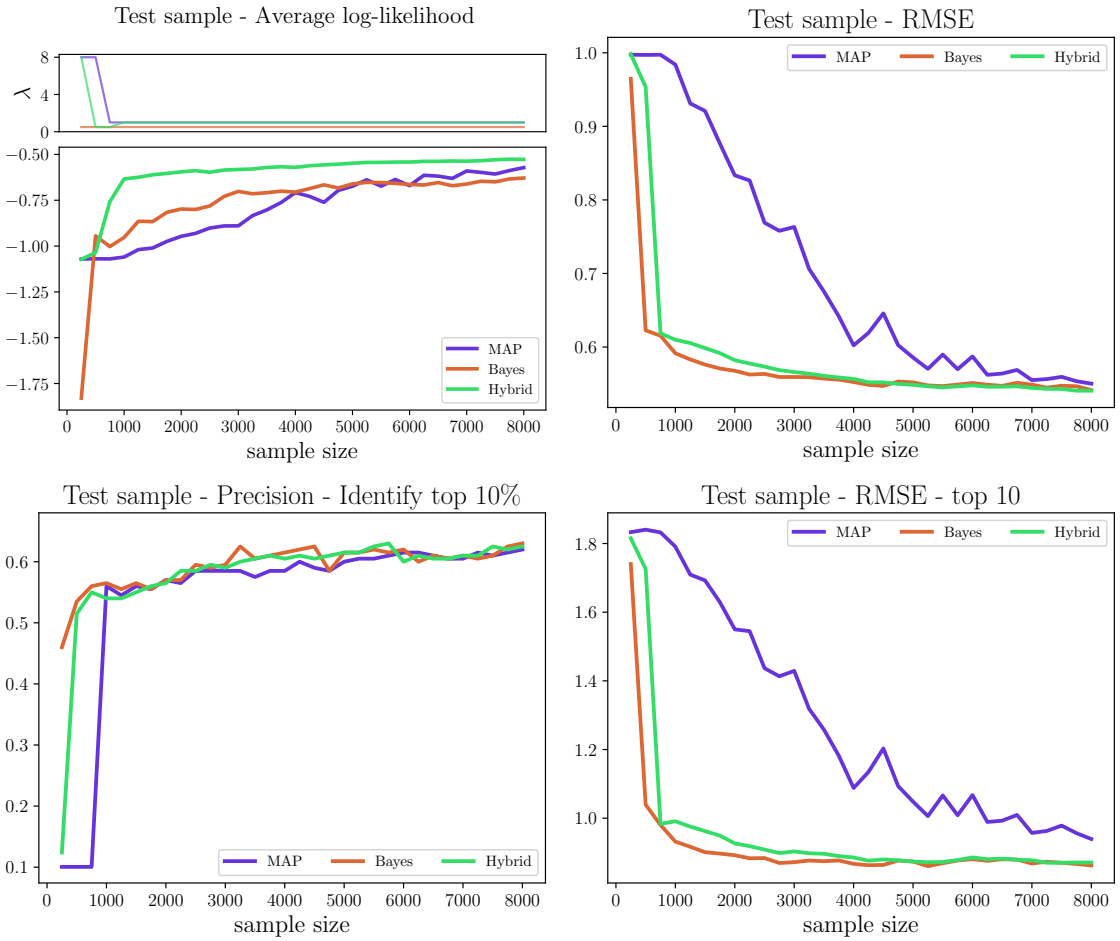

Figure 6 **Various measures of out-of-sample predictive performance for simulated samples ranging from 100 to 10,000 observations.**

We present the results in Figure 6. Interestingly, the the hybrid approach achieves the highest out of sample likelihood across most sample sizes, while the standard Bayesian and MAP performances are comparable. In terms of out-of-sample RMSE, both the standard and regularized Bayesian approaches vastly outperform MAP up to 5000 observations. The RMSE for the top 10% of prices shows a similar pattern with a slightly larger RMSE and the MAP performance being slower to converge. In terms of identifying the top 10%, all approaches achieve comparable performance. We suspect that the slight difference in results compared to the simulation comes from the data distribution. We however note that in both cases the Bayesian approach achieves lower prediction errors. Finally, we also fitted a linear model (a $[55, 2]$ neural network) which achieves at best a RMSE of 0.61 (13% higher) with the full sample, showing that substantial gains can be achieved with the use of neural networks.

5. Conclusion

As firms attempt to leverage more diverse sources of data in settings where data availability might be limited, a need grows for a flexible method providing both reliable predictions and uncertainty estimation. We address this need by proposing to adopt Bayesian neural networks within a probabilistic modeling framework, and show that they provide better estimates while allowing us to distinguish model uncertainty from outcome variability. We show with a simulation that our approach achieves lower prediction errors, mitigates bias and provides better uncertainty estimation in small samples and in regions of the data distribution with fewer observations. We then analyze a public dataset and confirm our ability to reduce prediction errors in small samples. Our results open the door to the use of deep learning methods in a broad range of marketing applications where data of unknown structure and limited in availability. The double advantage of reducing bias and more accurately assessing uncertainty allows managers to make better informed decisions, notably by taking risk into account. As the number of data sources and data types continues to grow, we believe our approach to provide a strong foundation for future research and the development of new methods adapted to specific applications.

Declarations

Funding and Competing Interests. The authors have no funding or competing interests to declare as per the Competing Interests Policy of the journal. No external funding was received and their only employment is with their respective university.

References

Baydin AG, Pearlmutter BA, Radul AA, Siskind JM (2017) Automatic differentiation in machine learning: a survey. *The Journal of Machine Learning Research* 18(1):5595–5637.

Buchholz A, Chopin N, Jacob PE (2021) Adaptive tuning of Hamiltonian Monte Carlo within sequential monte carlo. *Bayesian Analysis* 16(3):745–771.

Burda M, Daviet R (2022) Hamiltonian sequential Monte Carlo with application to consumer choice behavior. *Econometric Reviews* (Forthcoming).

Cooil B, Winer RS, Rados DL (1987) Cross-validation for prediction. *Journal of Marketing Research* 24(3):271–279.

D'Amour A, Heller K, Moldovan D, Adlam B, Alipanahi B, Beutel A, Chen C, Deaton J, Eisenstein J, Hoffman MD, et al. (2022) Underspecification presents challenges for credibility in modern machine learning. *The Journal of Machine Learning Research* 23(1):10237–10297.

Del Moral P, Doucet A, Jasra A (2006) Sequential monte carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 68(3):411–436.

Dietterich T (1995) Overfitting and undercomputing in machine learning. *ACM computing surveys (CSUR)* 27(3):326–327.

Guo C, Pleiss G, Sun Y, Weinberger KQ (2017) On calibration of modern neural networks. *International conference on machine learning*, 1321–1330 (PMLR).

He K, Zhang X, Ren S, Sun J (2015) Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *Proceedings of the IEEE international conference on computer vision*, 1026–1034.

Hein M, Andriushchenko M, Bitterwolf J (2019) Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 41–50.

Jiang H, Kim B, Guan M, Gupta M (2018) To trust or not to trust a classifier. *Advances in neural information processing systems* 31.

Karras T, Aittala M, Hellsten J, Laine S, Lehtinen J, Aila T (2020) Training generative adversarial networks with limited data. *Advances in Neural Information Processing Systems* 33:12104–12114.

Kearns M, Mansour Y, Ng AY, Ron D (1997) An experimental and theoretical comparison of model selection methods. *Machine Learning* 27(1):7–50.

Lewis L (2019) Data cleaning in Python: examples from cleaning Airbnb data. URL `https://towardsdatascience.com/predicting-airbnb-prices-with-deep-learning-part-1-how-to-clear`

Li H, Xu Z, Taylor G, Studer C, Goldstein T (2018) Visualizing the loss landscape of neural nets. *Advances in Neural Information Processing Systems*, 6389–6399.

Marquardt DW (1980) A critique of some ridge regression methods: Comment. *Journal of the American Statistical Association* 75(369):87–91.

Neal RM (1996) *Bayesian Learning for Neural Networks*, volume 118 of *Lecture Notes in Statistics* (Springer New York).

Neal RM (2012) MCMC using hamiltonian dynamics. *arXiv:1206.1901* .

Salman S, Liu X (2019) Overfitting mechanism and avoidance in deep neural networks. *arXiv preprint arXiv:1901.06566* .

Schaffer C (1993) Overfitting avoidance as bias. *Machine learning* 10(2):153–178.

Wang DB, Feng L, Zhang ML (2021) Rethinking calibration of deep neural networks: Do not be afraid of overconfidence. *Advances in Neural Information Processing Systems* 34.

Ying X (2019) An overview of overfitting and its solutions. *Journal of Physics: Conference Series*, volume 1168, 022022 (IOP Publishing).