Statistics Surveys
Vol. 20 (2026) 46–134
ISSN: 1935-7516
https://doi.org/10.1214/26-SS155

Online prediction for streaming observational data

Bertrand Clarke [1] **and Aleena Chanda**[2]

[1] *Department of Statistics, University of Nebraska-Lincoln, e-mail:* bclarke3@unl.edu

[2] *Department of Statistics, University of Nebraska-Lincoln, e-mail:* ach100@juniata.edu

Abstract: The automated collection of streaming observational data has become standard and defies most traditional analytic techniques. It is not just that models are hard to identify, there may not be any model that can be safely and usefully assumed. Indeed, frequently it is only predictions that can be made and assessed. Problems for this kind of data are often called $\mathcal{M}$-Open and have motivated new statistical approaches and philosophies.

This paper will review some of the most successful recent predictive methods for the $\mathcal{M}$-Open problem class. Techniques include predictors from Bayesian nonparametrics such as Gaussian process priors, predictors using expert advice such as the Shtarkov solution, hash function based predictors such as the Count-Min sketch, conformal prediction, and neural nets including Long Short-Term Memory networks.

MSC2020 subject classifications: Primary 62L99, 62C99; secondary 62M20.
Keywords and phrases: $\mathcal{M}$-Open, Shtarkov, Bayes prediction, hash functions, data stream algorithms, conformal prediction.

Received December 2024.

Contents

arXiv: 2507.21308v1

1. Introduction

The automated collection of large scale observational data sets has become commonplace. Often this goes by the name of webscraping, prediction along a string, or, as we call it here, streaming data – leaving it understood that we mean data less structured than time series.

For present purposes, we think of the paradigm streaming data set as high volume, continually and rapidly generated, and flowing endlessly i.e., as long as required with no meaningful beginning or end. The data is observational; we are not trying to intervene in any way. We have a data generator (DG) that simply runs. Our task is to process the stream of data outcome by outcome. That is, our processing is a stream, too. Since there is no meaningful stopping point we do not use batch processing. In particular, since the output for time n must be generated before the outcome for time $n+1$ is received, our analysis is in real time. Hence, our analytic procedures are time sensitive and incremental in the data. We do not have time to redo an analysis that we want to disavow. Another constraint is that we have limited data storage so while we can accumulate some data or some summaries of data we must discard most of it. Taken together, our analysis must be 'one pass' – we look at the new data point and our accumulated data summary, and then compute our output for the next time step in one running-time bounded procedure.

In particular, if we want to use a procedure that is not one-pass, we have to make it one pass. Here we do this by pre-processing the data. For instance, if use a clustering algorithm such as streaming K-means, we can simply fix a number of cluster centers K, use a burn-in of say $K' \gg K$ data points, and take the K cluster centers as a representative set for the data that has accumulated. Now, when the data run, i.e., we get data point $K'+1$, $K'+2$ and so on, the representative set will update according to streaming K-means and our procedure that was not one pass becomes one pass.

Perhaps surprisingly, there are many sources of streaming data. A few are: sensor data, financial data, weather data, real-time online purchases, the internet of things, image data, changepoint data, and much agricultural data. However, despite the profusion of techniques practitioners have for these settings, the development of general analytic techniques for streaming data seems to have lagged their demand. Technological progress has pressed the world of data collection, management, and analysis to leap ahead of what methodologists have developed to date.

On the other hand, sequential prediction has a long history in statistics even if most of it does not relate to streaming data. Perhaps the earliest prediction oriented treatment of statistics is Aitchison and Dunsmore (1975). It offered the first unified framework for prediction in what we now call $\mathcal{M}$-closed problems, see Subsec. 1.3.1. Prior to this book, predictive thinking was scattered as special cases across various subfields of statistics and applications – regression and early econometrics, for instance. Nearly 20 years later, Geisser (1993) began to persuade people that prediction should be the central organizing principle of Statistics in place of estimation, testing, and modeling more generally. His point was that predictive inference has always sought to focus on observables and uncertainty about the DG more than classical inference such as estimation and testing at the (high) cost of assuming well-defined models. Concurrent with the development of these ideas, Dawid (1984) enunciated the prequential principle as a contrast with the likelihood principle and the idea of revising models sequentially as data accumulate. Earlier, Dawid (1982) had noted that bias was undetectable from within the Bayes paradigm necessitating some form of empirical approach.

More recently the celebrated textbook Cesa-Bianchi and Lugosi (2006) provided a treatment of sequential prediction that was nearly encyclopedic up to the early 00's in particular for what we now call 'predictions with expert advice' and other standard predictors e.g., exponentially weighted average forecasts. This text has numerous specific examples of techniques with (often pointwise) performance bounds, e.g., on the 'regret' that we will see in Sec. 3. The results are often only for one predictive round at a time although some are cumulative.

1.1. The predictive view of data

Our focus on streaming data means we are effectively outside the classical modeling framework as exemplified, for instance, by Lehmann's two books Lehmann (1959) and Lehmann (1983). A key aspect of going 'beyond Lehmann' can be summarized by a paraphrase from Ransohoff (2005) in the context of complex DG's: All models are guilty of bias until proven innocent. That is, all models are intrinsically biased or at least oversimplified. Hence, we must identify the bias and account for it to ensure the reliability of our inferences. The key way to detect bias is via predictive performance. Moreover, it is important to recognize that in many $\mathcal{M}$-complete or -closed problems, bias arises mainly from oversimplification, even though this is often a necessary step in developing inferences.

By contrast, the traditional dictum 'all models are wrong but some are useful' has limited utility because as the complexity of the DG increases, our ability to detect and correct bias typically decreases. Indeed, complexity militates against interpretability. Often, our only way to assess the performance of predictors is with each other, ignoring the status of any belief, let alone true, model.

The situation for streaming and other complex data is actually worse than this. We are *assuming* there is not enough structure in the DG to allow effective modeling. In particular, there is nothing stable enough about the data stream for modeling to seize on reliably. For instance, if there are occasional jumps in the data we are assuming they are too variable to be modeled effectively. Otherwise put, whatever the application and terminology, the data are too complex to fit into a traditional probabilistic modeling framework. When data has fewer and weaker properties than we are accustomed to assuming, we should use techniques that make weaker assumptions – and are less prescriptive.

One can argue that if data are sufficiently chaotic, then probability modeling as an enterprise must be abandoned. Here, we largely regard this as a moot philosophical point and ignore it. Nevertheless, we will see that some sorts of probabilistic modeling are more effective than others. This suggests, but does not establish, that the role of probability theory remains substantial but of a different character than in conventional 'Lehmann derived' statistical analyses.

Admitting that the kind of streaming data we want to analyze is so non-representative of the data types we have traditionally tried to model means that for the most part we are thrown back on prediction and the properties of predictors such as robustness. For streaming decisions, we may also have an assessment of the costs of the decisions we have made, but we do not treat this case here. Our decisions will only be about what value we think the next outcome will take i.e., the n-th stage one-pass point prediction for the outcome at the $n+1$ time step.

One of the implications of our work is that, as a generality, Bayes methods work better than frequentist methods. The reason is simple: there is no probability model for the data. Hence, there is no meaningful basis to form a sampling distribution. By contrast, Bayes methods make weaker assumptions. In a posterior predictive, for instance, we need not assume there is a marginal distribution for the conditioning data and the distribution on the future outcome is only conceptual i.e., in our minds. While it depends on the past data via the mathematical operation of conditioning, there is no requirement that past data follow any distribution at all. The posterior can be seen simply as an input-output relationship, see Chen (1985). We develop this point in Sec. 2.

1.2. The techniques presented here: A look ahead

Here we present five established methods to form one-step-ahead point predictors that can be said to be specifically for streaming data. They will be discussed in roughly historical order.

The first technique is a stochastic process approach. Even though we rule out stochastic processes as a model for the DG, stochastic processes, if used well, and properly interpreted, can give philosophically valid point predictors. The main one developed here is the Gaussian process, or more exactly, since this class is mainly Bayesian, a Gaussian process prior (with a random bias) on a function space. When using this class we have to be careful to ensure posterior convergence does not put us back in an $\mathcal{M}$-complete or -closed case where there is a true model. In general, Bayes methods can be used with appropriate caveats.

For the sake of completeness, we also include a subsection on Bayesian time series. These are typically used for modeling for which reason we have chosen to downplay them here.[1] Time series models are varied and typically have frequentist and Bayes versions. Per the discussion in Sec. 2, Baysian time series can be philosophically consistent with streaming data. By contrast, the absence of a sampling distribution makes frequentist time series problematic.

The second technique is the Shtarkov solution. It rests on a finding a predictor that achieves the minimax regret under a log criterion – the regret is the

[1] Another problem with discussing even just Bayesian time series here is that the field is simply too vast to do it justice given that there are many excellent treatments of Bayesian time series, if not from an $\mathcal{M}$-open predictive standpoint.

difference between the best possible prediction that could have been made and the prediction that was actually made. This predictor is not used very much, often for computational reasons, but is conceptually important as a framework for thinking about sequential $\mathcal{M}$-open prediction. It emerged from the information theory literature in the late 80's and has both a Bayesian and a frequentist version, with the Bayesian version being generally better behaved. In this technique, the prior is put over a class of experts, not DG's, whose predictions are combined. Bayesians prefer to generate whole distributions for observables rather than just point predictors. However, it is unclear what this means for streaming $\mathcal{M}$-open data where there is no true distribution.

The third class of techniques originates from computer science and rests on randomly generated 'hash' functions. Hash functions are (usually discrete) functions that are not one-to-one – they reduce their domains – and so can be used for data compression. The name probably originates from its normal English usage – hash functions metaphorically chop and mix their domains to give a sort of uniformity on their range. In practice we have to choose several, perhaps many, hash functions but by allowing a small and controlled amount of error we can achieve good prediction and high computational efficiency.

The fourth class of techniques are conformal, the idea being that a future value should be in 'conformity' with the earlier values. This necessitates a measure of conformity and while not the same as a model such measures put all data points on the same scale in a heuristic sense. Performance assessment of conformal predictors is normally in $\mathcal{M}$-complete or -closed contexts but their empiricism makes them applicable in $\mathcal{M}$-open problems as well.

The fifth class of techniques are neural nets (NN's). NN's span a wide variety of predictive techniques, many relatively recent. Here we briefly discuss fully connected feedforward NN's as an introduction to recurrent NN's (RNN's) and Long Short-term Memory (LSTM) NN's. The latter are used in numerous predictive problems and typically involve explanatory variables. We have not included these here for simplicity: an elementary case without explanatory variables can be usefully defined. LSTM's do not have a necessary error structure although quantities like the mean squared error can be empirically defined.

We also present examples, results, and computational comparisons for these five methods. In particular, the goal of our computational work is only to compare the performance of the point predictors these methods generate in distance and stability senses. Indeed, even though we have required all our methods to be one-pass, the data sets we use here are not as large as we typically mean by the term streaming data. So, the one-pass constraint is only for predictive comparison purposes not practical computation. Despite these limitations, we identify methods that put more emphasis on the classes of predictors (as opposed to any properties of the data) as being better, heuristically, than methods that make assumptions on the DG – or appear to.

Since the only limit on the number of predictive techniques for streaming data is human imagination, we have had to omit many techniques that, on first blush, look promising. For instance, we have omitted treatments of kernel methods such as RVM's. Kernel methods are often based on the representer theorem and do not yield a classical model in the Lehmann sense: if used sequentially the number of terms (evaluations of the kernel function) can change. In particular, the number of terms can increase with n if there is no probabilistic mode in which to assess convergence as in $\mathcal{M}$-open problems. Also, we have omitted random forests. They are essentially bagged trees and are a fully nonparametric sequentially updated predictor. The energetic reader will have to pursue these and other predictive techniques elsewhere.

1.3. The class of problems studied here

We focus on point prediction in $\mathcal{M}$-open problems. To provide context for this, we discuss problem classes and the properties of prediction techniques.

1.3.1. The $\mathcal{M}$-open class of problems

We start to structure our thinking about sequential prediction by recalling the division of problems into three classes based on the relative positions of the data generator (DG) and the mathematical structures we have proposed to understand it. These originate in Bernardo and Smith (2000) and were called $\mathcal{M}$-closed, -completed, and -open. The idea is that we have a DG and what we will generically call a collection of 'models', or, better for present purposes, 'predictors'. The classes of problems depend on the relationship between the DG and the models/predictors. Here we update this tripartite partition of problem classes for our use in prediction as follows. We only do this briefly here, referring to Clarke and Yao (2025) for more details.

In the simplest case, called $\mathcal{M}$-closed, one of the models is true in the sense that it provides a complete and correct description of the DG. That is, the only unknown aspect of the problem is which 'model' in our class is true. It is understood that at least in an asymptotic sense, the true model gives the (unique) optimal predictor; see Rissanen (1984) Theorem 2 for an instance of this. Other inferences proceed from a chosen model, taking variability into account.

One step up in complexity is the $\mathcal{M}$-complete class of problems. In this problem class we assume there is a class of predictors and a DG but we do not assume any of the predictors are optimal even though they may be close. Since every valid scientific model must give a unique predictor, models can be included in the predictor class of an $\mathcal{M}$-complete problem. Thus, from the modeling standpoint, the true model which is assumed to exist, and would be available via its predictor, is not in the class. It is understood that, despite this, the resulting bias is not usually the issue. Instead, the intuition is that the true model and/or optimal predictor is inaccessible for some reason. So, the model or predictor class chosen is an approximation, hopefully a good one, at least in the sense that model bias is smaller than other sources of error.

Because there is an optimal predictor even if we cannot write it down explicitly, it can be transformed into a model, at least approximately. That is, it is possible to propose a sequence of models whose predictors converge to the optimal predictor. Consequently, expectations and convergences still exist and can still be used.

The most complex class of problems is called $\mathcal{M}$-open. In this case, we do not have a belief model and we are left to assess the elements of our model class based on the data we have at hand and nothing more. In this context, usually predictors and their properties are all that we have to examine. An easy way to think of this is that there is no model that can be usefully and safely identified. In the absence of a true model expectations, convergences, and evaluations based on anything more than the data are infeasible.

As *gedanken* examples of the three classes of predictive problems, consider the following. A data set generated a $N(\mu, 1)$ gives an $\mathcal{M}$-closed problem. A protein synthesis problem involving numerous biochemical inputs, catalysts, and other cellular conditions is very likely to be $\mathcal{M}$-complete since we may be convinced there is a model that can be approximated under various circumstances even if we cannot write it down. A possible example of an $\mathcal{M}$-open data set might be taking the world's great literature, converting it to a string of letters or characters, and then trying to produce i.e., predict, a new piece of great literature. In this latter example, predictors can be formed but there is likely no useful way to predict the next great novel.

In general, prediction in $\mathcal{M}$-open problems should satisfy the prequential principle. The basic statement is 'evaluations of predictors should be disjoint from their construction', see Dawid (1984). Henceforth, we impose this criterion and assume our data come from an $\mathcal{M}$-open DG.

1.3.2. Point prediction vs other predictions

Assume we have an $\mathcal{M}$-open DG that produces $\{y_1, \ldots, y_n, \ldots\}$ with $y_i \in \mathbb{R}$ for all i, i.e., the outcomes do not arise from a stochastic process. Since there is nothing repeated that we can estimate or test, our main task is responding to a collection of data $y^n = (y_1, \ldots, y_n)$ for any n in anticipation of receiving y_{n+1}. Essentially this is a sequential decision problem and here our decision will simply be a prediction for y_{n+1}.

What kind of prediction? Four options are: point predictors (PP), probabilistic predictions i.e., an entire predictive distribution (D), individual probabilities of events (P), and prediction sets that satisfies some coverage statement (S). Roughly, each method we consider here has a primary output in one of these categories. This is indicated in Table 1.

In addition, under the Prequential Principle, all of these methods are comparable in terms of the point predictors they generate. For instance, a probabilistic predictor can be used to generate a point predictor by various methods – simply take a mean, mode or median. Predictors that give individual probabilities can be combined into a distribution that again can generate a location predictor. If the output of a prediction scheme is a set, then in the real line, the endpoints can be averaged to give a point predictor. Hence, in our computational work we only use point predictors and look at some of their properties.

Table 1

List of the five conceptual classes of predictive methods and some of their key properties. LSTM's are non-stochastic models – unless a noise term is included or the recurrence is regarded as stochastic. Many of the entries in this table are subjective. For instance, in the Bayes row, one can argue that a prior amounts to putting a distance on a parameter space

Method	Inputs	Not required	Output
Bayes	prior, likelihood	distance	D
Shtarkov	log-loss, experts prior for Bayes	likelihood	D
Hash-based	partition, values	likelihood, distance	P
Conformal	conformity measure	prior, likelihood	S
LSTM	model	prior, likelihood, distance	PP

Table 1 is only heuristic but does provide a summary. The term likelihood refers only to a function of a parameter given the past data, not a conditional distribution. Distance means an actual metric or some object that functions like a metric such as a conformity or loss function (or score function). Prior really means some sort of probabilistic structure not over the data, not a Bayesian's actual prior. In fact, the hash functions are chosen probabilistically so hash-based methods have an implicit prior, even if it's always taken to be uniform. The term 'values' means that there are many tuning parameters that must be chosen. The term 'model' means a full parametric model must be specified.

Somewhat subjectively, looking at the assumptions of the methods, LSTM's make the strongest ones being a full model and giving only point predictors. Shtarkov looks to make the second strongest assumptions requiring a likelihood and a sense of distance here the log-loss (or possibly other score function) and a prior for the Bayes case. It's not clear which of conformal or Bayes has stronger assumptions. Both however give intervals. Hash-based methods make the weakest assumptions and only naturally give probabilities.

Looking at the methods from the standpoint of strength of inferences, arguably, the methods that produce a distribution – Bayes and Shtarkov – provide the strongest inferences, with Shtarkov making the strongest assumptions. However, it is unclear whether distributional inferences are meaningful in $\mathcal{M}$-open problems or not. On the other end, hash-based methods and LSTM's that only naturally give probabilities or point predictors provide the weakest inferences. Arguably, conformal methods are in the middle.

In principle, all of these methods allow multidimensional response variables and covariates. While this is obviously important, we do not address it here purely for simplicity.

1.4. Structure of this paper

The rest of this paper proceeds as follows. In Sec. 2, we adapt Bayesian thinking to $\mathcal{M}$-open DG's and present some basic techniques using Gaussian process priors and Dirichlet process priors. In Sec. 3 we present the minimax regret approach, focusing on the Shtarkov solution and predictor. In Sec. 4, we present the concept of hash functions as a technique to achieve good prediction, running time, and data storage properties simultaneously. Here we will explicitly use techniques, including imposing a 'one-pass' requirement, to ensure that our predictors scale up to high volume data. In Sec. 5 we briefly discuss conformal prediction and in Sec. 6 we describe neural networks and LSTM's in particular. We conclude with computational results in Sec. 7 including how we use streaming K-means to ensure non-one-pass predictors become one-pass. We provide a more general discussion of predictive methodologies in Sec. 8. Many of the routine proofs are relegated to Appendices A, B, C, and D.

2. Bayesian methods

Philosophically, Bayesian methods fit well with streaming data problems because Bayesians condition on past data and have a concept of updating upon receipt of more data. Moreover, as a matter of practice, they treat collected data as deterministic, i.e., as if they no longer have any stochastic properties. That is, the variability in the data is 'transmuted' into variability of an estimand or a future value Y_{n+1} that follows a conditional distribution given the data. So, strictly speaking, Bayesians only need a likelihood to form posterior quantities. It is desirable that the likelihood come from a model but this is not necessary, as will be explained shortly.

By contrast, frequentist inferences rest on the sampling distribution which is derived from a probability model for the DG. So, the frequentist regards data as outcomes of a random variable that can in principle be repeatedly sampled. In $\mathcal{M}$-open problems there is no probability model and hence no sampling distribution so it is unclear how frequentist methods can be applied at all.

Since Bayesians do not use a sampling distribution as a description for post-experimental variability, they do not have to put a distribution on the data. This is consistent with assuming the data are $\mathcal{M}$-open.

The Bayesian tendency to regard received data as fixed rests on the interpretation of conditional probability. Specifically, Chen (1985) presents results such as posterior normality to show that the convergence can be regarded as non-probabilistic. That is, to characterize the behavior of posterior quantities, it is enough if the data sets form a well-defined deterministic sequence rather than having any distributional properties. In essence, this means that Bayesian inference can be regarded as simply an input-output relation: deterministic data in, probabilistic inferences out. There is no contradiction in regarding the posterior distribution as simply a set function on the parameter space indexed by the past data regarded as deterministic inputs.

Quoting Chen (1985): 'our asymptotic results are non-probabilistic in nature; that is, the data sets $\mathcal{D}_n$ are not necessarily related across n, but simply form a well-defined deterministic sequence.' In short, much like our treatment of $\mathcal{M}$-open data in Sec. 3, Bayesian inferences only depend on having a string of data. No distributional properties of the sequence are required to characterize the behavior of posterior quantities. Chen (1985) also states that convergences in this deterministic sense can be improved to probabilistic convergences if the assumptions are taken as probabilistic as well. Informally, this suggests the likelihood is doing the work of convergence more than the data.

As a final philosophical point, one could regard frequentist procedures as an input-output relation from a string of data to a sampling distribution. The problem then is that this is just not consistent with the frequentist view that the likelihood is actually a probabilistic model under repeated sampling. There is no frequentist analog to Chen (1985).

To use this 'deterministic' property of Bayesian methods in a streaming context, ideally we would like to avoid convergence of, say, the posterior predictive to a limit under stochastic assumptions, IID being the simplest. The reason is that if the posterior predictive converges to a specific density e.g., $p(y_{n+1}|y^n) \rightarrow p(y_{n+1}|\theta_0)$ in some mode under some set of regularity conditions as n increases, it's as if we are saying that, even apart from the mode, the $\mathcal{M}$-open data eventually have a distribution and hence are not truly $\mathcal{M}$-Open. However, no theorem states that convergence of the posterior predictive to a limiting distribution implies that the data comes from that distribution – and conditions to ensure that sort of conclusion would necessarily exclude $\mathcal{M}$-open data. So, this is simply an intuitively reasonable desideratum. The converse is not obvious either, i.e., that the failure of a convergence such as $p(y_{n+1}|y^n) \rightarrow p(y_{n+1}|\theta_0)$ density does not imply a sequence of data is necessarily $\mathcal{M}$-open. On the other hand, one can argue that such convergences are frequentist and not relevant.

Given the foregoing, there are at least three ways to think about Bayesian prediction in streaming $\mathcal{M}$-Open problems that respect the lack of data distributions and possible concerns about convergences.

First, and easiest, is not to worry about it. Just let the data do it: if the data really are $\mathcal{M}$-open then there is no optimal predictor or 'true' distribution although one may be better than another. So, in principle, the posterior predictive, for instance, needn't converge anywhere and if it does, the convergence is irrelevant. As noted above, even if the posterior predictive did converge to a limit in some sense, this does not imply the data were generated by the limit. Relying on the data is a tidy approach to the problem, if perhaps unsatisfying.

Second, for streaming data where there is no replication, there is no loss in thinking of each $Y_i \sim p_i(y_i)$ for a collection of p_i's that are completely unrelated to each other. If the p_i's were chosen probabilistically, e.g., independent and tied together with a prior, the problem would become $\mathcal{M}$-complete: there is a true model but it is inaccessible to us and this doesn't matter because it is only a trivial constraint.

Since this sort of 'model' is unimplementable, it may be worthwhile to interpret it by a more effective approximation. One possibility, suggested by MacEachern (2023), is to regard each Y_i as having a random bias. That is, regard each Y_i as being of the form $Y_i + A_i$ where the A_i's form a sequence of independent random bias terms and we only observe the sum $y_i + a_i$. This is *de facto* $\mathcal{M}$-open because the constraint of the modeling is so minimal. Moreover, it will only be our analytic techniques that makes use of the random bias. The hope is that the flexibility added to the predictor by the bias will mimic the behavior of $\mathcal{M}$-open data thereby reducing the error. Otherwise put, reducing the inferential power of the predictor by including the random bias may 'fool' the predictor into being a better approximation of an optimal predictor for data that do not have a distribution.

Third, as an alternative to using a random bias, we can prevent the posterior distribution from concentrating at a limit point, if desired, by ensuring its variance does not go to zero. That is, with the interpretation from Chen (1985) if the posterior itself converges to a nontrivial distribution, or simply does not converge at all, then *a fortiori* the posterior predictive distribution will not concentrate to a member of the models used to form it. Indeed, even if the posterior, and hence posterior predictive, converges somewhere, it will not have the same meaning as in $\mathcal{M}$-complete or -closed cases. This alternative is difficult to formulate in practice even though it may be the most realistic. Hence, here we think about Bayes predictors in the $\mathcal{M}$-open context in the second way.

The remaining problems with Bayesian techniques are mainly computational. Standard Bayes theory advocates conditioning on all the data even though there is evidence this can be suboptimal predictively. Nevertheless, we follow this here and note further that as data accumulate, in some schemes like Gaussian process priors (GPP's), it is impossible to reduce the data to a small set of sufficient or nearly sufficient statistics and the number of parameters grows with n. Moreover, since running time is important, we will impose a 'one-pass' constraint; details on this are in Sec. 7.

In this section we will give three nonparametric Bayes predictors. One comes from using zero mean GPP's for function estimation; the second is similar but the GPP's has a mean given by a random bias term; and the third simply uses a Dirichlet process prior on distributions. All of these are relatively familiar but now they are being used in an $\mathcal{M}$-open context. We conclude this section with a brief discussion of Bayesian time series on the grounds that Bayes methods are applicable even if the actual time series models are unjustified.

2.1. Gaussian process priors

We start with our two GPP based predictors. The posterior predictive distribution for a new function value under a mean zero GPP is well-known and we simply quote it here. In the case that the GPP has a random additive bias we state what the posterior predictive density is and identify one way to obtain values for the hyperparameters to specify the predictor fully.

2.1.1. No bias

Given a stream of data Y_i, for $i = 1, \ldots, n$, the idea is to assume $Y_i = f_i + \epsilon_i, \quad i = 1, \cdots, n$ where the ϵ_i's are IID $N(0, \sigma^2)$. Now, the i^{th} data point y_i is an outcome of Y_i and we can equip $f = (f_1, f_2, \cdots, f_n)^T$ with a Gaussian process prior. That is,

$$f \sim \mathcal{N}(a, \sigma^2 K_{11}), \tag{1}$$

where $a = (a_1, a_2, \cdots, a_n)^T$ and $K_{11} = \left(\left(k_{ij}\right)\right)$ for $i, j = 1, \cdots, n$. In this subsection, we assume the means $a_i = 0$ for all i. So, letting $\epsilon = (\epsilon_1, \ldots, \epsilon_n)^T$, the joint distribution of $Y = (Y_1, Y_2, \cdots, Y_n)^T$ and Y_{n+1} is

$$\begin{aligned} \begin{pmatrix} Y \\ \vdots \\ Y_{n+1} \end{pmatrix} &= \begin{pmatrix} f \\ \vdots \\ f_{n+1} \end{pmatrix} + \begin{pmatrix} \epsilon \\ \vdots \\ \epsilon_{n+1} \end{pmatrix} \\ &\sim \mathcal{N}\left[\begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}, \sigma^2 \begin{pmatrix} K_{11} + I & \vdots & K_{12} \\ \cdots & \vdots & \cdots \\ K_{21} & \vdots & K_{22} + 1 \end{pmatrix} \right] \end{aligned} \tag{2}$$

where $K_{12} = (k_{1,n+1}, k_{2,n+1}, \cdots, k_{n,n})^T$, $K_{21} = K_{12}^T$, and $K_{22} = k_{n+1,n+1}$. Compactly, we write (2) as

$$Y^{n+1} \sim \mathcal{N}(0^{n+1}, \sigma^2 (I + K)_{n+1 \times n+1}). \tag{3}$$

It is well known from normal theory that the predictive distribution of Y_{n+1} given Y^n is the conditional

$$Y_{n+1} | Y^n \quad \sim \quad \mathcal{N}(\mu^*, \Sigma^*),$$

where

$$\mu^* \quad = \quad \sigma^2 K_{12} \{\sigma^2 (K_{11} + I)\}^{-1} y = K_{12} \{(K_{11} + I)\}^{-1} y \tag{4}$$

and

$$\begin{aligned} \Sigma^* &= \sigma^2 (K_{22} + 1) - K_{21} \{\sigma^2 (K_{11} + I)\}^{-1} \sigma^2 K_{12} \\ &= \sigma^2 (K_{22} + 1) - K_{21} (K_{11} + I)^{-1} K_{12}. \end{aligned} \tag{5}$$

In the zero bias case of all $a_i = 0$, the optimal point predictor (under squared error loss for instance) is simply the conditional mean μ^* in (4).

To complete the specification, it remains to estimate σ^2. From (3), using n in place of $n+1$ gives $Y^n \sim \mathcal{N}(0, \sigma^2 (I + K)_{n \times n})$. Hence, $(I + K)^{\frac{1}{2}} Y^n \sim \mathcal{N}(0, \sigma^2 I)$. Define $Y' = (I + K)^{\frac{1}{2}} Y^n$. Since the sample variance is an unbiased estimate of population variance, we estimate σ^2 by $S_2' = \frac{1}{n-1} \sum_{i=1}^n (y_i' - \bar{y'})^2$.

2.1.2. IID random additive bias

The suggestion for the technique in this subsection comes from MacEachern (2023). From (2), we see that

$$Y = Y^n \sim \mathcal{N}(a, \sigma^2(I_{n\times n} + K_{n\times n})).$$

So, the likelihood is given by

$$\begin{aligned}\mathcal{L}_1(a, \sigma^2|y) &= \mathcal{N}(a, \sigma^2(I_{n\times n} + K_{n\times n}))(y) \\ &= \frac{e^{-\frac{1}{2\sigma^2}(y-a)'(I_{n\times n}+K_{n\times n})^{-1}(y-a)}}{(2\pi)^{\frac{n}{2}}(\sigma^2)^{\frac{n}{2}}|I_{n\times n} + K_{n\times n}|^{\frac{1}{2}}}. \end{aligned} \tag{6}$$

It remains to choose priors for a and σ^2. First, we equip $a = (a_1, \ldots, a_n)^T$ with the distribution

$$a \sim \mathcal{N}(\gamma\mathbf{1}_n, \sigma^2\delta^2 I_{n\times n}), \tag{7}$$

where $\mathbf{1}_n$ denotes the vector of ones of length n. We treat γ as a hyperparameter representing the mean of the IID random bias in the Y_i's and use an empirical Bayes approach to obtain a serviceable value for it. Also, as discussed below, we treat $\delta^2 > 0$ as a deterministic parameter chosen for convenience. To complete the specification of the model we equip σ^2 with an inverse-Gamma distribution

$$\sigma^2 \sim \mathcal{IG}(\alpha, \beta). \tag{8}$$

Below, we will also use an empirical Bayes approach to find serviceable values for α and β. Thus, explicitly, our joint prior is

$$\begin{aligned} w(a, \sigma^2|\alpha, \beta, \gamma, \delta) &= \mathcal{N}(\gamma 1, \sigma^2\delta^2 I_{n\times n})\ \mathcal{IG}(\alpha, \beta) \\ = \frac{e^{-\frac{1}{2\sigma^2}(a-\gamma 1)'(\delta^2 I_{n\times n})^{-1}(a-\gamma 1)}}{(2\pi)^{\frac{n}{2}}(\sigma^2\delta^2)^{\frac{n}{2}}} & \frac{\beta^\alpha}{\Gamma(\alpha)} \times \left(\frac{1}{\sigma^2}\right)^{\alpha+1} e^{-\frac{\beta}{\sigma^2}}. \end{aligned} \tag{9}$$

The following theorem gives a predictive distribution for Y_{n+1}.

Theorem 2.1. *The posterior predictive distribution of a future observation Y_{n+1} given the past observations y^n is*

$$m(y_{n+1}|y^n) = St_{2\alpha+n}\left(A_1, \frac{\beta^*}{\frac{2\alpha+n}{2}}\right)(y_{n+1}). \tag{10}$$

In (10), $St_v(\theta, \Sigma)$ denotes the Student's t distribution with v degrees of freedom, location parameter θ and scale parameter Σ.

The form of β^* is given by $\beta^* = \beta + A_2$ and the form of A_1 is given by $A_1 = \frac{g_2-(y^n)^T g_1^n}{g_1}$ where, g_1^n, g_1, g_2, and A_2 can be explicitly written as functions of y^n, γ, δ, and the $(n+1)\times(n+1)$ variance matrix K.

For a proof of this result, see Chanda et al. (2026).

It is immediate from (10) that A_1 is the appropriate point predictor for Y_{n+1}. Indeed, the main effect of the random bias on point prediction is to make the location dependent on the data in a nonlinear way, cf. (4). Moreover, even though it may seem inconsistent to assign a distribution to Y_{n+1} when we assume it doesn't have one, the posterior predictive is only a statement of our belief for the location – not a statement about the actual behavior of Y_{n+1}.

To estimate α, β, γ, and δ we start with γ and adopt a maximum likelihood approach. We first write the joint likelihood of y^n and a^n given γ, δ^2 and σ^2. Then, we integrate out the a^n and write the result as product of a function of γ and a function of the other parameters that can be ignored. More explicitly, $\mathcal{L}(y^n, a^n|\sigma^2, \delta^2, \gamma)$ equals

$$\frac{1}{(2\pi)^{\frac{n}{2}}(\sigma^2)^{\frac{n}{2}}|I_{n\times n}+K_{n\times n}|^{\frac{1}{2}}} e^{-\frac{1}{2\sigma^2}(y^n-a^n)'(I_{n\times n}+K_{n\times n})^{-1}(y^n-a^n)}$$
$$\times \frac{1}{(2\pi)^{\frac{n}{2}}(\sigma^2\delta^2)^{\frac{n}{2}}} e^{-\frac{1}{2\sigma^2}(a^n-\gamma 1^n)'(\delta^2 I_{n\times n})^{-1}(a^n-\gamma 1^n)}.$$

Integrating out over a^n and re-arranging gives

$$\begin{aligned}\mathcal{L}_2(y^n|\gamma, \delta^2, \sigma^2) \quad &= \quad h(\gamma)\frac{|V_{n\times n}|^{\frac{1}{2}}}{(2\pi\sigma^2\delta^2)^{\frac{n}{2}}|(I+K)_{n\times n}|^{\frac{1}{2}}} \\ &\quad\times e^{-\frac{1}{2\sigma^2}\left[y'^{\,n}\left\{(I+K)^{-1}_{n\times n}+(I+K)^{-1}_{n\times n}V_{n\times n}(I+K)^{-1}_{n\times n}\right\}y^n\right]} \qquad (11)\end{aligned}$$

where

$$h(\gamma) \quad = \quad e^{-\frac{1}{2\sigma^2}\left[-2\gamma y'^{\,n}\frac{(I+K)^{-1}_{n\times n}V_{n\times n}}{\delta^2}1+\gamma^2 1'\left(\frac{I}{\delta^2}-\frac{V_{n\times n}}{\delta^4}\right)1^n\right]}$$

and the other factor on the right side of (11) can be discarded. Taking logarithms on both sides of h gives

$$\ln h(\gamma) = -\frac{1}{2\sigma^2}\left[-2\gamma y'^{\,n}\frac{(I+K)^{-1}_{n\times n}V_{n\times n}}{\delta^2}1^n + \frac{\gamma^2}{\delta^2}1'^{\,n}\left(I_{n\times n}-\frac{V_{n\times n}}{\delta^2}\right)1^n\right]. \quad (12)$$

Differentiating (12) with respect to γ, and equating it to zero leads to

$$\hat{\gamma} = \frac{y'^{\,n}(I+K)^{-1}_{n\times n}V_{n\times n}1^n}{1'^{\,n}\left(I_{n\times n}-\frac{V_{n\times n}}{\delta^2}\right)1^n}.$$

A second derivative argument gives that $\hat{\gamma}$ is typically a local minimum, at least for small $\delta > 0$; see Clarke and Chanda (2025).

Next, we find effective values for α and β. We do this for the sake of completeness because α and β are only required to identify the posterior predictive from Theorem 2.1; they are not needed to identify the point predictor A_1 for Y_{n+1}. Our procedure rests on a method of moments argument.

Recall from (6) that $Y \sim \mathcal{N}(a, \sigma^2(I+K)_{n\times n})$. Hence, $(I+K)^{\frac{1}{2}}Y \sim \mathcal{N}(a, \sigma^2 I)$. So, if we define $Y' = (I+K)^{\frac{1}{2}}Y$ we can set $S_2' = \frac{1}{n-1}\sum_{i=1}^n (y_i' - \bar{y'})^2$. We use the first two moments of S_2' to find values $\hat{\alpha}$ and $\hat{\beta}$ for given $\delta > 0$.[2] In fact,

$$\sigma^2 = \mathsf{E}\frac{S_2'}{1+\delta^2}, \tag{13}$$

see (90) in Appendix A.1; the argument giving (13) is different from the argument for S_2' at the end of Subsec. 2.1.1. So, for given $\delta > 0$, we use

$$\hat{\sigma}^2 = \frac{S_2'}{1+\delta^2}$$

in our first moment condition based on S_2', because it is unbiased.

Since we have two hyperparameters α and β, we need the second moment of $\hat{\sigma}^2$ as well. From (97), in Appendix A.1 we have that

$$\mathsf{Var}(\hat{\sigma}^2) \quad = \quad \frac{2\sigma^2}{(n-1)^2}\left[\sigma^2 + \frac{2n\gamma^2}{1+\delta^2}\right]. \tag{14}$$

So, we can approximate (14) as

$$\widehat{\mathsf{Var}}(\hat{\sigma}^2) = \frac{2\hat{\sigma}^2}{(n-1)^2}\left[\hat{\sigma}^2 + \frac{2n\gamma^2}{1+\delta^2}\right].$$

To finish this specification, recall (8). For the inverse-Gamma we have

$$\begin{aligned} \hat{\sigma}^2 \approx \mathsf{E}(\sigma^2) \quad &= \quad \frac{\beta}{\alpha-1} \\ \widehat{\mathsf{Var}}(\hat{\sigma}^2) \approx \mathsf{Var}(\sigma^2) \quad &= \quad \frac{\beta^2}{(\alpha-1)^2(\alpha-2)}. \end{aligned}$$

From these equations we can solve solve for α and β to find

$$\hat{\alpha} \approx \frac{\mathsf{E}^2(\hat{\sigma}^2)}{\mathsf{Var}(\hat{\sigma}^2)} + 2 \tag{15}$$

$$\hat{\beta} \approx \mathsf{E}(\hat{\sigma}^2)(\hat{\alpha} - 1). \tag{16}$$

Now, by plugging in estimates for the expectations we get values for $\hat{\alpha}$ and $\hat{\beta}$.

Finding a good value for δ is more problematic. One way to find a value would be by maximum likelihood: find a version of $\mathcal{L}$ analogous to (11), but with only one factor being a function of δ^2. That is, form the likelihood $\mathcal{L}_3(y|\gamma, \delta^2, \sigma^2)$ by integrating out a from the product of (6) and (7) and maximizing it over δ. Unfortunately, we cannot simply differentiate $g(\delta^2)$, set the derivative to zero, and solve. The resulting equations are just too complicated to be useful.

[2] For contrast, at the end of Subsec. 2.1.1, we could estimate σ by using $\hat{\sigma}^2 = S_2'$ because $\sigma^2 = ES_2'$. This does not hold here because of the random bias.

In principle, one could do a grid search over δ^2 in an interval $\mathbf{I} \subset \mathbb{R}^+$ to maximize $\mathcal{L}_3(y|\gamma, \delta^2, \sigma^2)$. However, in computational work not shown here, we found that the optimal $\delta \in \mathbf{I}$ was almost always the left hand end point, even as $\mathbf{I}$ moved closer and closer to 0. Hence, pragmatically, in our computations here, we simply chose δ to be small. Note that if $\delta = 0$, the prior on a is degenerate and in fact $\gamma = 0$ as well. This means that the mean and variance of our bias a is zero i.e., there is no bias. Practical choices of δ in our examples were around .1 so that variability in the prior would not overwhelm the data. To compensate, we did robustness analyses to verify the stability of our results using values of δ as large as 1.

2.1.3. INID random additive bias

For comparison with the earlier two examples using GPP's, consider the case where the distribution of a^n is independent but not identical. i.e.,

$$a^n \sim \mathcal{N}(\gamma^n, \sigma^2\delta^2 I_{n\times n}), \tag{17}$$

where $\gamma^n = (\gamma_1, \ldots, \gamma_n)^T \in \mathbb{R}^n$. This means that at the $n+1$ stage when we want to predict Y_{n+1}, we have an extra location parameter γ_{n+1} in addition to the first n parameters in γ^n. Theorem 2.1 can be formally extended to this case. We have the following.

Theorem 2.2. *The posterior predictive distribution of the future observation Y_{n+1} given the past observations y^n is*

$$m(y_{n+1}|y^n) = St_{2\alpha+n}\left(A_1^*, \frac{\beta^{**}}{\frac{2\alpha+n}{2}}\right)(y_{n+1}). \tag{18}$$

In (18), the form of β^{**} is given by $\beta^{**} = \beta + A_2^*$ and the form of A_1^* is given by $A_1^* = \frac{g_2^* - y'^n g_1^{*n}}{g_1^*}$ where g_1^{n*}, g_1^*, g_2^*, and A_2^* can be explicitly written as functions of y^n, γ^{n+1}, δ, and the $(n+1)\times(n+1)$ variance matrix K.

Proof. A proof of this result follows by replacing γ in the proof of Theorem 2.1 with γ^n and doing a line-by-line verification. A sketch of the proof can be found in Appendix A.2. □

It is problematic to use Theorem 2.2 in practice. To see the impediments, consider finding values of the parameters.

Suppose we seek a value for γ^n, the first n biases. If we use the analog of (11) from the IID case for the present INID case, we get a more general form of the function h. Taking logarithms on both sides of h gives an analog of (12):

$$\ln h(\gamma^n) \quad = \quad -\frac{1}{2\sigma^2}\left[-\frac{2}{\delta^2}y'^n(I+K)^{-1}_{n\times n}V_{n\times n}\gamma^n + \gamma'^n\left(\frac{I_{n\times n}}{\delta^2} - \frac{V_{n\times n}}{\delta^4}\right)\gamma^n\right].$$

Differentiating this w.r.t. γ^n and setting the derivative to zero gives

$$\hat{\gamma}^n = \left(I_{n\times n} - \frac{V_{n\times n}}{\delta^2}\right)^{-1}(I+K)^{-1}_{n\times n}V_{n\times n}y^n. \tag{19}$$

A second derivative argument, essentially the same as before, gives that $\hat{\gamma}^n$ is again a maximum.

However, if we replace n by $n+1$ in (19) we see that we need y_{n+1} to estimate γ_{n+1} or a value of γ_{n+1} to predict y_{n+1} and both are unknown. Hence, the non-identical biases make it impossible to predict y_{n+1}. One way around this is to assign a value to γ_{n+1} by some other technique; we have only explored this in one case, namely taking $\gamma_{n+1} = \bar{\gamma}_n$.

A few further comments about the form of A_1^* and β^{**}. First, A_1^* involves γ^{n+1} only because of g_2^*; g_1^{n*} and g_1^* do not involve γ^{n+1}. Second, only g_2^* and A_2^* involve γ^{n+1}. Third, similar to Subsubsec. 2.1.2, we can use a method of moments argument to find $\hat{\alpha}$ and $\hat{\beta}$. The details are given in Appendix A.3. Thus, even though we have a form for the predictive distribution of $Y_{n+1}|y^n$, in practice we are not able to use it without extra information.

2.2. *Dirichlet process priors*

Suppose a discrete prior G is distributed according to a Dirichlet Process (DPP), and write $G \sim DP(\alpha, G_0)$ where α is the mass concentration parameter (that we take to be one) and G_0 is the base measure with $\mathbb{E}(F) = G_0$. Then, by construction, we have the following standard results; see Ghosal (2010).

If the sample space $\mathbb{R}$ is partitioned into $A_1, A_2, \cdots, A_k$, the random vector of probabilities $G(A_1), G(A_2), \cdots, G(A_k)$ has a Dirichlet distribution, i.e.,

$$p(G(A_1), G(A_2), \cdots, G(A_k)) \sim \mathit{Dirichlet}(\alpha(A_1), \alpha(A_2), \cdots, \alpha(A_k)),$$

where $\alpha(\mathbb{R}) = M$, which we take here to be one.

By conjugacy, the posterior distribution of

$$G(A_1), G(A_2), \cdots, G(A_k)|Y_1, Y_2, \cdots, Y_n$$

is also Dirichlet but with parameters $\alpha(A_j) + n_j$ where

$$n_j = \sum_{i=1}^{n} I(Y_i \in A_j); j = 1, 2, \ldots, k.$$

If $Y_j'; j = 1, 2, \cdots, k$ are the distinct observations in $\{Y_i; i = 1, 2, \cdots, n\}$, the posterior predictive distribution of $Y_{n+1}|Y_1, Y_2, \cdots, Y_n$ is

$$Y_{n+1}|Y_1, Y_2, \cdots, Y_n = \begin{cases} \delta_{Y_j'}, \text{with probability} \\ \frac{n_j}{M+n}; j = 1, 2, \cdots, \text{k; and} \\ F_0, \text{with probability } \frac{M}{M+n} \end{cases} .$$

Now, our DPP predictor is

$$\hat{Y}_{n+1} \quad = \quad \sum_{j=1}^{k} y_j' \frac{n_j}{M+n} + \frac{M}{M+n} \mathsf{median}(F_0). \tag{20}$$

2.3. Time series

Very loosely, Bayesian predictions from time series methods stem from one of two approaches: Box-Jenkins models and state space models. Either can be multivariate but here we limit attention to univariate cases.

2.3.1. Bayesian Box-Jenkins

In the simplest formulation of the streaming data problem we have a stochastic process $Y_1, Y_2, \ldots$ in which relationships among the Y_i's are given in terms of the backshift operator B that acts on the individual random variables to give their one step earlier version, that is, $B(Y_{n+1}) = Y_n$. Often polynomials in B are used such as $\phi(B) = 1 - \phi_1 B - \cdots - \phi_p B^p$ where p is called the order of ϕ and for convenience we write $\phi = (\phi_1, \ldots, \phi_p)$ for the parameters. Now an auto-regressive moving average model is written as

$$\phi(B)Y_n = \theta(B)\epsilon_n \tag{21}$$

for each n in which θ is a polynomial of order q in B analogous to ϕ and the ϵ_n' are IID $N(0, \sigma^2)$. It is straightforward to put a prior on ϕ, θ, and σ: Using (21), write ϵ_n as the difference between Y_n and polynomials ϕ and θ in B acting on Y_n:

$$\epsilon_n = Y_n - \sum_{j=1}^{p} \phi_j Y_{n-j} - \sum_{j=1}^{q} \theta_j \epsilon_{n-j}. \tag{22}$$

Since the ϵ_i's are IID, using (22) the likelihood is

$$\mathcal{L}(\phi, \theta, \sigma | Y^n = y^n) \propto \left(\frac{1}{\sigma^2}\right)^{(n-p)/2} e^{\frac{1}{2\sigma^2} \sum_{t=p+1}^{n} \epsilon_t^2}.$$

Equipping ϕ, θ, and σ with priors, and assuming an initial distribution for Y_0 (that we have ignored here for simplicity) leads to the posterior $w(\phi, \theta, \sigma \mid y^n)$ and therefore to a posterior predictive $p(y_{n+1}|y^n)$. This process can be used to give a predictive density for multiple Y_i's and for more complicated models that introduce seasonality and 'integration' – a differencing of the time series separated by say d time points. The effect of these is to make the polynomials in (21) more complicated. Point predictors for $\hat{Y}_{n+1}$ can be derived from $p(y_{n+1}|y^n)$ by taking the posterior mean, for instance, and give PI's by using the posterior variance. Implementing these techniques involves computational issues that are beyond our present scope.

2.3.2. Bayesian state space models

A state space model (SSM) is a generalization of a Box-Jenkins model that represents the observation at time i, i.e., Y_i, as a function of an underlying state

X_i with $\dim(X_i) = d$ that satisfies a transition equation. With initial state X_0, the sequence of random variables is

$$X_0 \longrightarrow \begin{pmatrix} X_1 \\ Y_1 \end{pmatrix} \rightarrow \begin{pmatrix} X_2 \\ Y_2 \end{pmatrix} \longrightarrow \cdots \begin{pmatrix} X_{n+1} \\ Y_{n+1} \end{pmatrix} \tag{23}$$

and it is implicitly assumed that Y_i is available after X_i and X_{i+1} is available after Y_i. Typically, it is also assumed the X_i's are Markov and the 'transition' equation can be written $X_i = f(X_{i-1}, \eta_i)$ for some f, where the η_i's are IID noise. Also, the 'observation' equation can be written $Y_i = g(X_i, \epsilon_i)$ for some g, where the ϵ_i's are also IID. Thus, an SSM is, at root, a hidden Markov process.

One reason (23) is called an SSM is that one of the main tasks is to identify the underlying 'state' $X_i = x_i$ from the observations y_1^n. When f and g are linear and the noise terms η_i and ϵ_i are normal, the set of updating equations (mean and variance) for X_{n+1} given Y^n is often called Kalman filter prediction; the corresponding set of expressions for X_{n+1} given Y^{n+1} is often called Kalman filter updating. The term 'filter' refers to the fact that as n increases, the conditioning σ-fields $\sigma(Y^n)$ or $\sigma(Y^{n+1})$ increase. The analysis of (23) is 'Bayesian' in that Kalman filters for prediction or updating look like posteriors for X_{n+1} given past observations $Y^n = y^n$ or $Y^{n+1} = y^{n+1}$.

To be more precise, we follow the derivation in Gurajala et al. (2021) for Kalman filter prediction. For a more detailed and equally lucid treatment, see Petris et al. (2009) Chap.2. Let $\eta_i \sim N(0, Q)$ and $\epsilon_i \sim N(0, R)$ and fix matrices F, G and H so that

$$\begin{aligned} X_{n+1} &= FX_n + G\eta_n \\ Y_{n+1} &= HX_n + \epsilon_n. \end{aligned}$$

This structure is often called dynamic linear regression, especially when $F = F_n$, $G = G_n$ and $H = H_n$. Since all variables are normal and all transformations are linear, it is easy to derive a point predictor for $(X_{n+1}|Y^n = y^n)$ and its variance as

$$\begin{aligned} \hat{X}_{n+1} &= FE(X_{n+1}|Y^n = y^n) \\ \mathrm{Var}(X_{n+1}|Y^n = y^n) &= F\mathrm{Var}(X_n|Y^n = y^n)F^\top + GQG. \end{aligned}$$

Similar expressions hold for Kalman filter updating. The standard assessment of Kalman filter convergence is how well the PI's formed from mean and covariance match the data over time. When the various parameters are accurate, $\hat{X}_{n+1}$ becomes a linear function and $\mathrm{Var}(X_{n+1}|Y^n = y^n)$ goes to a constant at rate $\mathcal{O}(1/n)$. Given the expressions for Kalman filter prediction, the predictive density for Y_{n+1} is

$$p(y_{n+1}|y^n) = \int p(y_{n+1}|x_{n+1})p(x_{n+1}|y^n)\mathrm{d}x_{n+1}$$

and the conditional density for $(X_{n+1}|X^n)$ is just the transition equation.

In a fully Bayesian treatment F, G, H, Q, and R would be regarded as parameters equipped with a prior. Such a treatment would lead to a more flexible model with intuitive inferences. Bayesian models also allow specification of non-Gaussian state disturbances and observation innovations. However, this more general treatment does not seem to have been carried out, perhaps due to sample size considerations. As with Bayesian Box-Jenkins, we have not included SSM's in our computational work because these are usually regarded as models – not $\mathcal{M}$-open – and are beyond our present scope.

3. Shtarkov

The Shtarkov solution, see Shtarkov (1987), is a density that can be used for prediction under a minimax regret optimality criterion. The criterion assumes that there are multiple experts each giving predictions for a future outcome and that a forecaster may use these to form the actual prediction that will be announced. The forecaster's goal is only to match the predictive performance of the best expert. In practice, the 'experts' are represented by their candidate models that represent their honest views of what the next outcome will be.

Often this is phrased as a game of n rounds between a Forecaster and Nature, overseen by an MC. At the start of the first round, the MC calls for each expert to announce a predictor for the first outcome y_1, here assumed univariate. The experts are indexed by θ and are represented by predictive densities say $p(y|\theta)$, for $\theta \in \Theta$ where Θ is the set of experts. For ease of exposition, we assume Θ is a connected open set in d-dimensional real space and that $p(y|\theta)$ is continuous and bounded on Θ in both its arguments.

Next, the Forecaster uses these $p(y_1|\theta)$'s to form a predictive density $q(y_1)$ that will be used to generate the actual predictions for the next outcome y_1. Seeing all this, Nature generates the actual outcome y_1 by any mechanism at all – probabilistic or not. Thus, the prediction task is $\mathcal{M}$-Open. The MC compares the prediction and the outcome by computing $q(y_1)$ and instructing the Forecaster to pay Nature $\ln(1/q(y_1))$, concluding the round; when $\ln(1/q(y_1))$ is negative it is the amount Nature pays the Forecaster. We assume that each player wants to make as much money (or lose as little money) as possible.

If n rounds of the game are played sequentially, then Forecaster's predictor $q(y_i)$ for outcome y_i, $i = 2, \ldots, n$ may use the earlier observed y_i's. Also, the experts may use the earlier data to form the predictors they announce. The Forecaster only uses the earlier data by way of the experts' predictors.

This game uses ln as if it were a loss function. In fact, we are using ln as a scoring rule: it measures how representative an outcome is of a density. Values of y with higher values of $q(y)$ are more representative (of q) than values of y with lower values of $q(y)$. There are many scoring rules. They have different properties and forecasts are sensitive to which one is chosen.

A physical interpretation comes from information theory. For discrete outcomes, the values $\ln(1/q(y))$ are approximately the Shannon code lengths when q is the source distribution and therefore it is a measure of complexity – higher complexity y's correspond to longer code words and lower probabilities. When the base of the logarithm is two, code length is a cost because it is the number of bits that must be used in a binary representation of the outcome. An extensive discussion of the ln scoring rule, the role of experts, and sequential prediction more generally is in Cesa-Bianchi and Lugosi (2006).

The question is how the Forecaster should choose q and make use of the experts' predictors. The Forecaster may decide that, rather than form predictions independently based on no information, it may be best to 'follow the best expert'. That is, consistently choose the predictor from the expert θ who gives the best predictions. This is formalized in the concept of regret: regret is the loss the forecaster incurs beyond the loss of the best performing expert. The best performing expert depends on the data and may vary from round to round.

The difference between the Forecaster's prediction and an expert's density under the ln scoring rule is

$$\mathsf{Regret}(q,\theta,y) = \ln\frac{1}{q(y)} - \ln\frac{1}{p(y|\theta)} = \ln\frac{p(y|\theta)}{q(y)}. \tag{24}$$

The maximal regret is

$$\mathsf{R}_{\mathsf{max}}(q) = \sup_y \sup_\theta \ln\frac{p(y|\theta)}{q(y)}. \tag{25}$$

The optimal $q(\cdot)$ minimizes (25) and is given by the Shtarkov solution

$$q_{\mathsf{opt,F}}(y) = \arg_q \left[\min_q \sup_y \sup_\theta \ln\frac{p(y|\theta)}{q(y)}\right] = \frac{p(y|\hat{\theta})}{\int p(y|\hat{\theta})dy}, \tag{26}$$

where $\hat{\theta} = \theta(\hat{y})$ is the maximal likelihood estimate (MLE), provided the integral exists. If θ is discrete, the integral becomes a sum that must be finite. So, Nature can maximize the cost to the Forecaster by choosing

$$y = \arg\max_y \; \ln 1/(q_{\mathsf{opt,F}}(y))$$

when the maximum exists, provided Nature knows the Forecaster will use $q_{\mathsf{opt,F}}$. The expression in brackets in (26) is the minimax regret, so $q_{\mathsf{opt,F}}$ is the minimax predictor that we call the Shtarkov solution. A proof of (26) is in Shtarkov (1987), see also Barron et al. (1998), exp. (5).

There is a Bayesian analog of (24) and (25). Suppose $w(\theta)$ is a prior density that we interpret as representing the credibility of the expert θ or the pre-prediction worth of the action defined by $p(\cdot|\theta)$. Now our regret becomes

$$\mathsf{Regret}_{\mathsf{B}}(q,\theta,y) = \ln\frac{1}{q(y)} - \ln\frac{1}{w(\theta)p(y|\theta)} = \ln\frac{w(\theta)p(y|\theta)}{q(y)}. \tag{27}$$

The maximal regret becomes

$$\mathsf{R}_{\mathsf{B,max}}(q) = \sup_y \sup_\theta \ln\frac{w(\theta)p(y|\theta)}{q(y)} \tag{28}$$

and the optimal $q(\cdot)$ that minimizes (28) is the Bayes Shtarkov solution

$$q_{\mathsf{opt,B}}(y) = \frac{w(\tilde{\theta}(y^n))p(y^n|\tilde{\theta}(y^n))}{\sum_{y^n} w(\tilde{\theta}(y^n))p(y^n|\tilde{\theta}(y^n))}, \tag{29}$$

where $\tilde{\theta}$ is the maximum posterior likelihood estimator (MPLE), i.e., the posterior mode. In (29), we have written the solution for discrete θ; if θ were continuous, we would replace the sum with an integral cf. (26).

In $\mathcal{M}$-open problems the status of w as a prior is unclear but it can be regarded simply as a pre-data preference for some experts over others, perhaps in the sense of reliability. In this view, the experts are essentially regarded as analogous to actions in a decision theory problem rather than as distributions. Without further comment, we take w to be continuous. For each n, the Shtarkov solution gives a density often called the normalized maximum likelihood (NML). In the Bayes case these are normalized maximum posterior likelihoods (NMPL). For convenience, we write q_{opt} when a statement applies to both $q_{\mathsf{opt,F}}$ and $q_{\mathsf{opt,B}}$.

These quantities, (26) and (29), have been studied extensively. In particular, Grünwald (2007) studies both the frequentist and Bayes cases extensively, using the term 'luckiness' where we have used the term Bayes. Chaps. 6-11 develops sequential prediction theory thoroughly including the introduction of a conditional form of (26) i.e., conditioning on a small fraction of the data, to ensure normalizing constants are finite. Later chapters e.g., Chap. 11, use the term luckiness where we have used the term Bayes and prior, see p. 308-9 and especially 'LNML-II' p. 311. Separately, the recent treatment in Yamanishi (2023) is particularly lucid, see Chap. 2.5 (Bayes) and Chap. 5.1 (frequentist). The former also refers to the prior in (29) as a 'luckiness' function, The Bayes/luckiness version originates in Clarke and Dawid (1998) from a different motivation but only appears in Clarke (2007).

An interesting feature of the Shtarkov solution is that it is almost outside the usual purview of probability theory. Recall, the Kolmogorov Extension Theorem identifies two conditions that are sufficient for a set of distributions to form a stochastic process; see Øksendal (2003), p. 11. One of these is a marginalization condition and it is reasonable to conjecture that q_{opt} does not satisfy it. So, a natural question is when the sequence of NML or NMPL predictive densities, the latter usually with the Jeffreys prior, forms a stochastic process. This question is largely resolved in a pair of papers, Hedayati and Bartlett (2012) and Bartlett (1975), that deserve to be better known. Taken together, these authors show (i) (29) and (26) asymptotically coincide and are optimal if and only if the latter is exchangeable and (ii) there are only three one-dimensional parametric families that have exchangeable NML densities (Gaussian, Gamma, and Tweedie exponential of order 3/2). Note that the statements are asymptotic. Even though Shtarkov and Bayes Shtarkov predictors are ultimately equivalent, for finite n the Bayesian's mixture distribution is suboptimal to both Shtarkov predictors, cf. Barron et al. (1998) and Clarke (2007).

A continual question is how to choose the experts since in practice we rarely have a bank of experts who will announce densities. When experts are available, they usually only give their predictions, not predictors for general use. In fact, the experts here are simply a parametric family and we are free to choose it however we wish. Regarding the $p(\cdot|\theta)$'s as announced by experts is simply an interpretation to ensure that our method makes sense in an $\mathcal{M}$-open context; we don't have to assume a parametric family is true to get useful results.

The Shtarkov solution can be used to give predictions. First, because it is a density we can form highest density prediction regions given a desired level of predictive confidence $1-\alpha$. However, since we don't assume the underlying DG is a probability we don't really believe the spread of q_{opt}. So, we only use q_{opt} to give point predictors. Often, the mode of the Shtarkov solution is used as a point predictor because q_{opt} can be very highly skewed (when it is not symmetric and unimodal); see Le and Clarke (2016) and the examples below.

Here, we use the maximum of

$$q_{\mathsf{opt,F}}(y^n, y_{n+1}) \quad \text{or} \quad q_{\mathsf{opt,B}}(y^n, y_{n+1})$$

over y_{n+1} holding y^n fixed, and write the predictions as $\hat{y}_{n+1,\mathsf{F}}(y^n)$ or $\hat{y}_{n+1,\mathsf{B}}(y^n)$. We call these the frequentist and Bayes Shtarkov predictors, respectively. This would be equivalent to finding the maximum of

$$q_{\mathsf{Sht}}(y_{n+1}|y^n) = \frac{q_{\mathsf{opt}}(y^{n+1})}{q_{\mathsf{opt}}(y^n)}$$

provided that the conditional density is well-defined. Since we must assume that the normalizing constants in $q_{\mathsf{opt}}(y^{n+1})$ and $q_{\mathsf{opt}}(y^{n+1})$ exist, it is equivalent to find

$$\arg\max_{y_{n+1}} p(y^n, y_{n+1}|\hat{\theta}(y^{n+1})).$$

The next subsection looks at sufficient conditions for the NML and NPML to exist. Then we turn to some illustrative examples. Outside well behaved parametric families of experts few Shtarkov predictors can be given in closed form. However, as a generality, they can be given numerically, e.g., in the binomial case below. We conclude this section with a brief discussion of these and other issues in Shtarkov predictors.

3.1. Existence of $q_{opt,F}$

The main impediment to using the NML or NMPL as a density is ensuring it exists. Since results ensuring the existence of MLE's and MPLE's under mild conditions are well-known, the task here is to find conditions under which the denominators in (26) and (29) are finite. Similar problems can arise if other optimal predictors are used, see Cesa-Bianchi and Lugosi (2006). In fact, the denominator of the NML or NMPL can be infinite making the solution undefined. For instance, the NML constant for the $\mathsf{Exponential}(\lambda)$ is infinite because it is the integral of

$$p(y^n|\hat{\lambda}) = \frac{1}{\bar{y}}e^{-n} \tag{30}$$

with respect to $\mathrm{d}y_1 \ldots \mathrm{d}y_n$ over $(\mathbb{R}^+)^n$, where $\hat{\lambda} = 1/\bar{y}$. However, it is easy to see that the NML of the $\mathsf{Binomial}(N, p)$ is finite because it has finite support.

In some cases, an asymptotic result from Rissanen (1996) can be used. Effectively it gives

$$q_{\mathsf{opt,F}}(y^n) = \frac{d}{2} \ln \frac{n}{2\pi} + C + o(1) \tag{31}$$

and identifies C; using essentially the same proof gives an analogous result for $q_{\mathsf{opt,B}}$. One limitation of this approach is that it can be difficult to determine whether the conditions for its validity are satisfied. Another limitation is that n must be large so finding the exact value of the constant may be difficult if that is desired. Moreover, the conditions that lead to expressions like (31) are too strong: we don't need a nice asymptotic expression; we only want existence or, at most, useful finite sample approximations.

Some authors try to avoid the nonexistence of the NML and/or NMPL or the ineffectiveness of asymptotics by focusing on the Bayesian mixture distribution as an approximation for finite samples, see Barron et al. (2014) and Clarke (2007).

Here, we denote the normalizing constants in the NML and NMPL by

$$D_{n,\mathsf{F}} = \int p(y^n|\hat{\theta})dy^n \quad \text{and} \quad D_{n,\mathsf{B}} = \int w(\tilde{\theta})p(y^n|\tilde{\theta})dy^n.$$

There are two types of conditions that ensure the two forms of D_n exist as bounded, strictly positive real numbers. The first gives a result stronger than is actually needed: it gives a rate for the normalization constant as well as existence. Even though its conditions are mild, they are hard to verify. The second type of conditions are easy to check but are stronger than required.

The first type of conditions are based on Rissanen (1996), see also Le and Clarke (2016). With some informality, we have the following.

Theorem 3.1. *Assume the following:*

1. *Let $I_n(\theta)$ be the n^{th} stage Fisher Information and suppose there is an $I(\theta)$ so that*

$$I_n(\theta) \quad = \quad -\frac{1}{n} E\left[\frac{\partial^2 \ln p(y^n|\theta)}{\partial\theta_i \partial\theta_j}\right] \to I(\theta),$$

as $n \to \infty$, and suppose $\exists\, c_1, c_2$ so that $\forall \theta \in \Theta$, $0 < c_1 \le |I(\theta)| \le c_2 < \infty$.

2. *The elements of $I(\theta)$ are continuous on Θ.*
3. $$\int_\Theta \sqrt{I(\theta)}\mathrm{d}\theta < \infty.$$
4. *The posterior mode $\tilde{\theta}$ and the MLE $\hat{\theta}$ satisfy a uniform central limit theorem. That is, for $\tilde{\theta}$ we have*

$$\xi \quad = \quad \sqrt{n}(\tilde{\theta}(y^n) - \theta) \xrightarrow{\mathcal{L}} \mathcal{N}(0, I^{-1}(\theta))$$

uniformly for $\theta \in \Theta$ and similarly for $\hat{\theta}$.

5. *There is a positive definite matrix C_0 so that*

$$I(y^n, \tilde{\theta}) \quad = \quad \left(-\frac{1}{n} \left\{ \frac{\partial^2 \ln p(y^n|\theta)}{\partial\theta_i \partial\theta_j} \right\}_{\theta=\tilde{\theta}} \right)_{i,j=1,\cdots,k} < C_0 < \infty$$

assuming $\tilde{\theta} \in \Theta$, and similarly for $\hat{\theta}$. In addition, the family

$$I_{ij}(y^n, \theta(\xi)) \quad = \quad -\frac{1}{n} \frac{\partial^2 \ln p(y^n|\theta(\xi))}{\partial\theta_i \partial\theta_j},$$

where $\theta(\xi) = \tilde{\theta} + \xi/\sqrt{n}$ is equicontinuous at $\xi = 0$ for $n \geq 1$, $1 \leq i$, $j \leq k$ and similarly for $\hat{\theta}$.

Then, $D_{n,\mathsf{F}}$ and $D_{n,\mathsf{B}}$ exist and hence so do $q_{\mathsf{opt},\mathsf{F}}(\cdot)$ and $q_{\mathsf{opt},\mathsf{B}}(\cdot)$, respectively.

Proof. This follows from an examination of the proofs in Rissanen (1996) and Le and Clarke (2016). The convergences assumed in the conditions necessitate a restriction to sequences y^n for which $\hat{\theta}$ and $\tilde{\theta}$ are in the parameter space Θ. □

The second type of result has a much simpler argument, partially explaining why its conditions are stronger albeit easier to check.

Theorem 3.2. *Assume the conditions:*

1. *The parameter space is bounded, convex, and the interior of its closure.*
2. *There is an N so that for $n > N$ the likelihood function, resp. the joint density for the parameter and data, is continuous and strictly convex in θ.*
3. *The densities $\{p(y|\theta)\}$ have common bounded support in y as θ varies.*
4. *The densities $\{p(y|\theta)\}$ are continuous as real valued functions of two real vector valued arguments y and θ.*
5. *The prior density $w(\cdot)$ is positive and continuous on the parameter space.*

Then, $D_{n,\mathsf{F}}$ and $D_{n,\mathsf{B}}$ exist and hence so do $q_{\mathsf{opt},\mathsf{F}}(\cdot)$ and $q_{\mathsf{opt},\mathsf{B}}(\cdot)$, respectively.

Remark. Although the densities and random variables are assumed bounded, we conjecture that this can be relaxed. In particular, we think Condition 3 can be improved by using Theorem 2.1 in Mäkeläinen et al. (1981).

Proof. This is a straightforward exercise in real analysis, see Appendix B. □

The examples in the next subsection – Normal, Binomial, and Gamma – probably satisfy the conditions of Theorem 3.1. However, there is no easy and obvious way to verify Conditions 4 and 5 (and for the criterion on sets that was omitted from the statement for simplicity). On the other hand, it is easy to see that the binomial satisfies the conditions of Theorem 3.2 and that the other examples satisfy these conditions for compact subsets of the parameter space and truncations of the random variables.

3.2. Examples

In this subsection we start by presenting the Bayes and frequentist Shtarkov point predictors for the normal family of experts with both μ and σ unknown. Then we turn to the Bayesian binomial. Predictors in this case cannot be worked out in closed form. So, we approximate them computationally. The frequentist binomial is similar. For breadth, we also give predictors for the Gamma family. These can be worked out in closed form and are intuitively reasonable. It is important to recall that the 'experts' are not models; they are only predictors with no necessary physical correlates in terms of the DG.

3.2.1. Normal distribution

Consider normal experts with μ and σ^2 unknown. We have the MLE's $\hat{\mu}_n = \bar{y}_n$ and $\hat{\sigma}_n^2 = \frac{1}{n}\sum_{i=1}^n (y_i - \bar{y}_n)^2$. So,

$$\begin{aligned} p(y^n|\hat{\mu}_n, \hat{\sigma}_n^2) &= \frac{n^{\frac{n}{2}} e^{-\frac{n}{2}}}{(2\pi)^{\frac{n}{2}} [\sum_{i=1}^n (y_i - \bar{y}_n)^2]^{\frac{n}{2}}} \\ &\propto \frac{1}{[\sum_{i=1}^n (y_i - \bar{y}_n)^2]^{\frac{n}{2}}}. \end{aligned}$$

Writing this for $n+1$ and taking logarithms, we get

$$\begin{aligned} &\ln p(y^{n+1}|\hat{\mu}_{n+1}, \hat{\sigma}_{n+1}^2) \\ &\quad \propto -\frac{n+1}{2} \ln \sum_{i=1}^{n+1} (y_i - \bar{y}_{n+1})^2 \\ &\quad = -\frac{n+1}{2} \ln \left[\sum_{i=1}^n y_i^2 + y_{n+1}^2 - (n+1) \left(\frac{n\bar{y}_n + y_{n+1}}{n+1} \right)^2 \right]. \qquad (32) \end{aligned}$$

Differentiating with respect to y_{n+1}, setting the derivative equal to zero, and re-arranging gives

$$\hat{y}_{n+1} - \frac{1}{n+1}(n\bar{y}_n + \hat{y}_{n+1}) = 0$$

Solving this gives $\hat{y}_{n+1} = \bar{y}_n$; see Appendix C.1.

Next we want to ensure the existence of the Shtarkov solution. Consider the conditions in Theorem 3.1. First, Condition 3 does not hold because the parameter space is $\mathbb{R} \times \mathbb{R}^+$ and the Fisher information matrix is

$$I(\mu, \sigma) = \begin{pmatrix} 1/\sigma^2 & 0 \\ 0 & 1/2\sigma^4 \end{pmatrix},$$

so the integral over μ diverges and the theorem fails. On the plus side, Condition 2 holds and satisfying the asymptotic conditions 1, 4, and 5 requires finding regularity conditions that guarantee them. This can probably be done, but requires work, cf. Dudley (1999). On the other hand, it is easy to see that the conditions of Theorem 3.2 can be satisfied easily if we assume that (θ, σ) is restricted to a compact set and Y is assumed bounded by, say, $B < \infty$. That is, even though the actual Shtarkov solution does not exist, approximations using compact sets increasing to the entire parameter space $\mathbb{R} \times \mathbb{R}^+$ and $B \nearrow \infty$ may exist and may lead to the same predictors as if the actual Shtarkov solution existed.

More generally, one can argue that the optimization we use to find $\hat{y}_{n+1}$ is meaningful as long as the numerator in the NML or NMPL exists. Indeed, as we have defined it here, $\hat{y}_{n+1}$ is a sort of maximized likelihood – but treating y_{n+1} as the parameter. This is analogous to getting meaningful estimators or predictions from Bayes methods using improper priors because in both cases the problem is a divergent integral.

Extending this to the Bayes Shtarkov normal case where both μ and σ^2 are unknown, we use the standard priors

$$p(\mu|\mu_0, \sigma_0^2) = \left(\frac{1}{\sigma^2\sigma_0^2 2\pi}\right)^{\frac{1}{2}} e^{-\frac{1}{2\sigma^2\sigma_0^2}(\mu-\mu_0)^2} \quad \text{and} \tag{33}$$

$$p(\sigma^2|\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)}\left(\frac{1}{\sigma^2}\right)^{\alpha+1} e^{-\frac{\beta}{\sigma^2}}. \tag{34}$$

So, the joint likelihood and prior is

$$\begin{aligned} & p(y^n|\mu, \sigma^2) \times p(\mu|\mu_0, \sigma_0^2) \times p(\sigma^2|\alpha, \beta) \\ & \propto \left(\frac{1}{\sigma^2}\right)^{\frac{n}{2}+\alpha+1}\left(\frac{1}{\sigma^2}\right)^{\frac{1}{2}} e^{-\frac{1}{\sigma^2}[\beta+\frac{1}{2}\sum_{i=1}^n(y_i-\mu)^2+\frac{1}{2}(\frac{\mu-\mu_0}{\sigma_0})^2]}, \end{aligned} \tag{35}$$

and it is straightforward to see that

$$\begin{aligned} \mu|\sigma^2, \sigma_0^2, \mu_0 &\sim \mathcal{N}\left(\frac{n\bar{y} + \frac{\mu_0}{\sigma_0^2}}{n + \frac{1}{\sigma_0^2}}, \frac{\sigma^2\sigma_0^2}{n\sigma_0^2 + 1}\right) \quad \text{and} \\ \sigma^2|\alpha, \beta &\sim \mathcal{IG}\left(\alpha + \frac{n}{2}, \beta + \frac{1}{2}\left\{\sum_{i=1}^n y_i^2 + \frac{\mu_0^2}{\sigma_0^2} - \frac{(n\bar{y} + \frac{\mu_0}{\sigma_0^2})^2}{n + \frac{1}{\sigma_0^2}}\right\}\right). \end{aligned}$$

Thus, we have

$$\hat{\mu}_{\mathsf{MPLE}} = \frac{n\bar{y} + \frac{\mu_0}{\sigma_0^2}}{n + \frac{1}{\sigma_0^2}} \tag{36}$$

$$\hat{\sigma}^2_{\mathsf{MPLE}} = \frac{\beta + \frac{1}{2}\left\{\sum_{i=1}^n y_i^2 + \frac{\mu_0^2}{\sigma_0^2} - \frac{(n\bar{y}+\frac{\mu_0}{\sigma_0^2})^2}{n+\frac{1}{\sigma_0^2}}\right\}}{\alpha + \frac{n}{2} + 1}. \tag{37}$$

The conditions of Theorem 3.2 are not satisfied because they are too strong. As stated, they require a compact parameter space, although this can in principle be relaxed. However, in the present example with continuous parameters, it is easy to see that the denominator of the Bayes Shtarkov solution is finite, cf. (29) for the discrete parameter case.

For present purposes, it is enough to consider the special case in which $\mu_0 = 0$ and $\sigma_0^2 = 1$. Substituting these values into (36) and (37) we get

$$\begin{aligned} \hat{\mu}_{\mathsf{MPLE}} &= \frac{n\bar{y}}{n+1} && (38) \\ \hat{\sigma}^2_{\mathsf{MPLE}} &= \frac{\beta + \frac{1}{2}\left\{\sum_{i=1}^n y_i^2 - \frac{(n\bar{y})^2}{n+1}\right\}}{\alpha + \frac{n}{2} + 1}. && (39) \end{aligned}$$

Substituting (38) and (39) in (35) gives

$$\begin{aligned} p(y^n|\alpha,\beta) &\propto \left[\frac{\alpha + \frac{n}{2} + 1}{\beta + \frac{1}{2}\left\{\sum_{i=1}^n y_i^2 - \frac{(n\bar{y})^2}{n+1}\right\}}\right]^{\frac{2\alpha+3+n}{2}} \\ &\quad \times e^{-\frac{1}{2}\frac{\alpha+\frac{n}{2}+1}{\beta+\frac{1}{2}\left\{\sum_{i=1}^n y_i^2 - \frac{(n\bar{y})^2}{n+1}\right\}}\left[\beta + \frac{1}{2}\sum_{i=1}^n (y_i - \frac{n\bar{y}}{n+1})^2 + \frac{1}{2}\frac{n^2\bar{y}^2}{(n+1)^2}\right]} \\ &= \left[\frac{\alpha + \frac{n}{2} + 1}{\beta + \frac{1}{2}\left\{\sum_{i=1}^n y_i^2 - \frac{(n\bar{y})^2}{n+1}\right\}}\right]^{\frac{2\alpha+3+n}{2}} \times e^{-\frac{\alpha+\frac{n}{2}+1}{2}} \\ &\propto \left[\frac{1}{\beta + \frac{1}{2}\sum_{i=1}^n (y_i - \bar{y})^2 + \frac{1}{2}\frac{n}{n+1}\bar{y}^2}\right]^{\frac{2\alpha+3+n}{2}}. && (40) \end{aligned}$$

We show that the integral of (40) with respect to y^n over $\mathbb{R}^n$ is finite. Consider the transformation

$$\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} = \begin{pmatrix} \frac{1}{\sqrt{n(n+1)}} & \frac{1}{\sqrt{n(n+1)}} & \frac{1}{\sqrt{n(n+1)}} & \cdots & \frac{1}{\sqrt{n(n+1)}} & \frac{1}{\sqrt{n(n+1)}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} & 0 & \cdots & 0 & 0 \\ \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{6}} & \frac{-2}{\sqrt{6}} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{1}{\sqrt{n(n-1)}} & \frac{1}{\sqrt{n(n-1)}} & \frac{1}{\sqrt{n(n-1)}} & \cdots & \frac{1}{\sqrt{n(n-1)}} & \frac{-(n-1)}{\sqrt{n(n-1)}} \end{pmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}. \quad (41)$$

Write P for the $n \times n$ matrix in (41) and let $|P|$ be its determinant. Now,

$$\begin{aligned} \frac{n}{n+1}\bar{y}^2 &= u_1^2 && (42) \\ \sum_{i=1}^n (y_i - \bar{y})^2 &= u_2^2 + u_3^2 + \cdots + u_n^2 && (43) \\ \mathrm{d}y_1 \cdots \mathrm{d}y_n &= \frac{1}{|P|}\mathrm{d}u_1 \cdots \mathrm{d}u_n. && (44) \end{aligned}$$

Substituting (42), (43) and (44) into (40), we get

$$\begin{aligned} p(u^n|\alpha,\beta) &\propto \left[\frac{1}{\beta+\frac{1}{2}\sum_{i=1}^n u_i^2}\right]^{\frac{2\alpha+3+n}{2}} \frac{1}{|P|}\mathrm{d}u_1\cdots\mathrm{d}u_n \\ &\propto \left[\frac{1}{1+\frac{1}{2\beta}u^{n\prime}u^n}\right]^{\frac{2\alpha+3+n}{2}} \mathrm{d}u_1\cdots\mathrm{d}u_n. \end{aligned} \qquad (45)$$

Recall that the pdf for a multivariate t is of the form

$$St_v(\tau,\Sigma)(g) = \frac{\Gamma(\frac{v+d}{2})}{\Gamma(\frac{v}{2})\pi^{\frac{d}{2}}|v\Sigma|^{\frac{1}{2}}}\Big(1+\frac{(g-\tau)'\Sigma^{-1}(g-\tau)}{v}\Big)^{-\frac{v+d}{2}}. \qquad (46)$$

Let $v=2\alpha+3$, $d=n$, $\Sigma=\frac{\beta}{\frac{2\alpha+3}{2}}I_{n\times n}$, $\tau=0$, and $g=u$. Using these specifications in (46) we get

$$p(u^n|\alpha,\beta) \propto \frac{\Gamma(\frac{2\alpha+3}{2})\pi^{\frac{1}{2}}\times\beta^{\frac{1}{2}}}{\Gamma(\frac{2\alpha+3+n}{2})} \times St_{2\alpha+3}\left(0^n, \frac{\beta}{\frac{2\alpha+3}{2}}I_{n\times n}\right). \qquad (47)$$

Thus, we recognize that the normalizing constant in $p(y^n|\alpha,\beta)$ is finite for $\mu_0=0$ and $\sigma_0=1$. Moreover, different values of μ_0 and σ_0 won't change tail behavior and it's tail behavior that determines whether an integral is finite. Thus, in general, the normalizing constant in the two-parameter Bayes Shtarkov solution exists. This bound is implicit in Grünwald (2007); see the reasoning on p. 367 for $\mathcal{S}^X$ in Chap. 12.4.3. Outside the normal likelihood case with conjugate priors, one can, in general, choose a prior with tight enough tails that the Bayes Shtarkov solutions exists.

Finally, some calculus arguments, see Appendix C.1, lead to

$$p(y^{n+1}|\hat{\mu}_{n+1,\mathsf{MPLE}},\hat{\sigma^2}_{n+1,\mathsf{MPLE}}) \times p(\hat{\mu}_{n+1,\mathsf{MPLE}}|\mu_0,\sigma_0^2)p(\hat{\sigma}^2_{n+1,\mathsf{MPLE}}|\alpha,\beta)$$

being maximized for given y^n at the MPLE, see (36).

3.2.2. Binomial distribution

As a second example, we look at the binomial. Let $Y\sim\mathrm{Bin}(N,\theta)$ have probability mass function denoted

$$p(y|\theta)=\binom{N}{y}\theta^y(1-\theta)^{N-y} \qquad (48)$$

for $0\le\theta\le1$ and $y=0,1,2,\cdots,N$. For the frequentist case, recall that for n samples $y_1,y_2,\cdots,y_n$, the MLE of θ is

$$\hat{\theta}_{\mathsf{MLE}}=\frac{\sum_{i=1}^n y_i}{Nn}=\frac{\bar{y}_n}{N}. \qquad (49)$$

We have that for fixed N, $\binom{N}{y}\theta^y(1-\theta)^{N-y}$ is bounded as a function of y and θ. Thus, $\sum p(y^n|\hat{\theta}) = \sum_{y_i=0}^{N} \prod_{i=1}^{n} \binom{N}{y}\hat{\theta}^y(1-\hat{\theta})^{N-y}$ is bounded so the frequentist Shtarkov solution exists by Theorem 3.2. Moreover, for any well-behaved prior, a similar observation gives that the Bayes Shtarkov solution exists, too. These statements are unusually easy to obtain for the binomial because it has bounded support and a bounded parameter space. This reasoning holds for the multinomial, the Beta with a bounded parameter space, and many other examples.

For brevity, we only present the Bayes Shtarkov solution here. Recall, the log-likelihood function of $\theta|y_1, y_2, \cdots, y_n$ is

$$p(y^n|\theta) \quad = \quad \left\{\prod_{i=1}^{n}\binom{N}{y_i}\right\}\theta^{\sum_{i=1}^n y_i}(1-\theta)^{Nn-\sum_{i=1}^n y_i}. \tag{50}$$

The conjugate prior for θ is given by

$$w(\theta|\alpha,\beta) \quad = \quad \frac{1}{\mathcal{B}(\alpha,\beta)}\theta^{\alpha-1}(1-\theta)^{\beta-1}; \alpha>0, \beta>0, \tag{51}$$

where the Beta function is $\mathcal{B}(\alpha,\beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$ in terms of the Gamma function. Then, from Appendix C.2, we see that

$$\begin{aligned} &\ln[p(y^{n+1}|\hat{\theta}_{\mathsf{MPLE}})w(\hat{\theta}_{\mathsf{MPLE}},\alpha,\beta)] \\ &\propto \ln\binom{N}{y_{n+1}} + (n\bar{y}_n + y_{n+1} + \alpha - 1)\ln(n\bar{y}_n + y_{n+1} + \alpha - 1) \\ &+ (N(n+1) + \beta - (n\bar{y}_n + y_{n+1}) - 1)\ln(\beta + N(n+1) - n\bar{y}_n - y_{n+1} - 1). \end{aligned} \tag{52}$$

Clearly, there is no general closed form solution for $\hat{y}_{n+1}$ in terms of $\bar{y}$.

To get around this, we generate numerical solutions. Fig. 1 shows a plot of the log-likelihood in (52) for $N = 30$ and $n = 10$. The range of y_{n+1} is $\{0, 1, \ldots, 30\}$. The actual range of $\bar{y}_{n+1}$ is the same, but we used .5 to 30 in increments of .5 to get a smoother plot. (We omitted $\bar{y} = 0$ to ensure logarithms can be well-defined.)

For fixed $\bar{y}$, the curve in the surface of Fig. 1 is concave so we can maximize $\hat{y}_{n+1}$ uniquely over the range of $\bar{y}_n$. It is seen that for $\bar{y}$ near zero, the maximum is achieved by $y_{n+1} = 0$ and for $\bar{y} = 30$, the maximum is achieved by $y_{n+1} = 30$. The optimal values of y_{n+1} increase with $\bar{y}$.

Surprisingly, as n increases the maximum for a given $\bar{y}$ does not become stronger. For instance, Fig. 2 shows the plots of the maximized loglikelihood in (52) for for $n = 10$ and 25 using green and yellow dots respectively. The green dots are the values in the line $\bar{y} = 10$ in the surface plotted in Fig. 1 and the yellow dots are the same but for $n = 25$. The two dotted curves in Fig. 2 are nearly identical; the curvature does not increase with n.

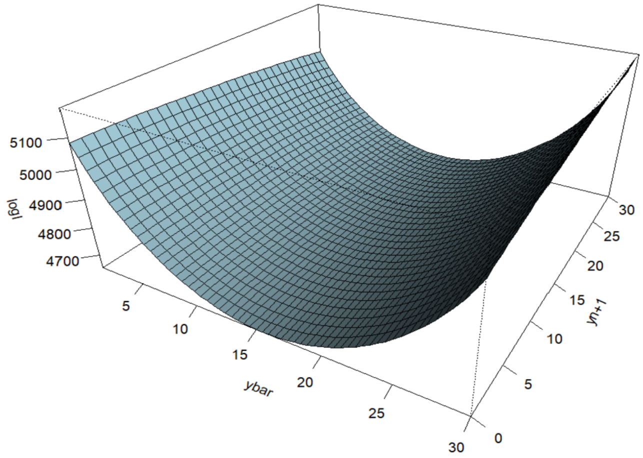

Fig 1. *Perspective plot of* $\bar{y}$ *and* y_{n+1} *from* (52) *for* $N = 30$ *and* $n = 10$*. The optimal* $\hat{y}_{n+1}$ *for a given* $\bar{y}$ *is the unique maximum in the surface over a line with a fixed value* $\bar{y}$*. Note that as* $\bar{y}$ *moves from zero to 30 the location of the maximum moves from zero to 30.*

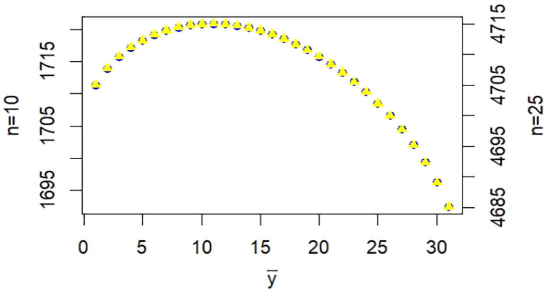

Fig 2. *For* $\bar{y}_n = 10$ *we plot the loglikelihood for* y_{n+1}*. The maximum at about 11-12 indicates optimal value for* $\hat{y}_{n+1}$*. The blue dots are for sample size* $n = 10$ *and the yellow dots are for sample size* $n = 25$*. It is seen that they are equal to within the resolution of the image file.*

3.2.3. Gamma distributions

The $Gamma(\alpha, \theta)$ is a generalization of the exponential model given by

$$p_\alpha(y^n|\theta) \quad = \quad \frac{\theta^{n\alpha}}{\{\Gamma(\alpha)^n\}}\left\{\prod_{i=1}^{n} y_i^{\alpha-1}\right\}e^{-\theta\sum_{i=1}^{n} y_i}. \tag{53}$$

For any fixed $\alpha > 0$, to predict Y_{n+1} we can find the Shtarkov predictor $\hat{y}_{n+1}$ optimizing over θ in the frequentist case. The impediment to optimizing over α as well is that differentiation brings in the digamma functions that are difficult to use. The Bayes case has the same problem and working with the conjugate prior for the joint parameter (α, θ) is also difficult. So, here, we treat α only as an index not as a parameter.

The existence of the Shtarkov solution in this example follows the same reasoning as in Subsec. 3.2.1 for the normal.

In the frequentist case, taking logarithms on both sides of (53) gives

$$\ln p_\alpha(y^n|\theta) \quad = \quad n\alpha \ln\theta - \theta\sum_{i=1}^{n} y_i + \ln\frac{\{\prod_{i=1}^{n} y_i^{\alpha-1}\}}{\Gamma(\alpha)^n\}}. \tag{54}$$

The MLE is $\hat{\theta}_{\mathsf{MLE}} = \alpha/\bar{y}_n$, Using this for $n+1$ outcomes gives

$$\begin{aligned} p_\alpha(y^{n+1}|\hat{\theta}_{\mathsf{MLE}}) &= \left(\frac{\alpha}{\bar{y}_{n+1}}\right)^{(n+1)\alpha} e^{-\frac{\alpha}{\bar{y}_{n+1}}(n+1)\bar{y}_{n+1}} \left\{\prod_{i=1}^{n} y_i^{\alpha-1}\right\} y_{n+1}^{\alpha-1} \\ &\propto \frac{y_{n+1}^{\alpha-1}}{(n\bar{y}_n + y_{n+1})^{(n+1)\alpha}}. \end{aligned} \tag{55}$$

This leads to

$$\hat{y}_{n+1} \quad = \quad \frac{n(\alpha-1)\bar{y}_n}{n\alpha+1} \tag{56}$$

as the form of our Shtarkov predictor for $\alpha \geq 1$, see Appendix C.3. For $\alpha \leq 1$ we simply get the Shtarkov predictor for the exponential family. It is identically zero which makes sense because most of the probability piles up around zero.

Turning to the Bayes case, let us choose the prior for θ to be

$$w(\theta|\alpha_0, \beta_0) \quad = \quad \frac{\beta_0^{\alpha_0}}{\Gamma(\alpha_0)}\theta^{\alpha_0-1}e^{-\beta_0\theta}. \tag{57}$$

Then the joint prior and likelihood is

$$\begin{aligned} p(y^n|\theta, \alpha) \times w(\theta|\alpha_0, \beta_0) &= \frac{\beta_0^{\alpha_0}}{\Gamma(\alpha_0)}\prod_{i=1}^{n} y_i^{\alpha-1}\frac{1}{\Gamma(\alpha)^n}\theta^{n\alpha+\alpha_0-1}e^{-\theta(n\bar{y}_n+\beta_0)} \\ &\propto \theta^{n\alpha+\alpha_0-1}e^{-\theta(n\bar{y}_n+\beta_0)}. \end{aligned} \tag{58}$$

So, the posterior distribution of $\theta|y^n, \alpha, \alpha_0, \beta_0$ is a Gamma distribution with parameters $(n\alpha+\alpha_0, n\bar{y}_n+\beta_0)$. Hence,

$$\hat{\theta}_{\mathsf{MPLE}} \quad = \quad \frac{n\alpha+\alpha_0-1}{n\bar{y}_n+\beta_0}. \tag{59}$$

Now, differentiating the log joint prior and likelihood, setting equal to zero and re-arranging gives

$$\hat{y}_{n+1} \quad = \quad \frac{(\alpha-1)(\beta_0+n\bar{y}_n)}{n\alpha+\alpha_0}. \tag{60}$$

Now, (i) for $\alpha > 1$, (58) is maximized by (60); (ii) for $\alpha = 1$, (58) is maximized by $\hat{y}_{n+1} = 0$; and (iii) and for $\alpha < 1$ (58) is maximized by $\hat{y}_{n+1} = 0$ provided n is large enough; see Appendix C.3. As in the exponential family, an identically zero predictor makes sense given the shape of the experts' distributions.

3.3. Other cases

The examples worked out here for the binomial generalize to the multinomial. Indeed, a lot of work has been done in this case because the computing becomes much more difficult especially outside the case of conjugate priors. For details and further work, see Cesa-Bianchi and Lugosi (2006) Chap. 3, Roos (2008) Kontkanen and Myllymäki (2007), and Le and Clarke (2016).

Computing Shtarkov solutions has attracted a fair amount of attention because of its coding applications. Rissanen (1996) offers a simple asymptotically valid formula under conditions not too different from Theorem 3. By contrast, citeBarron:etal:2014 focuses on computing the Shtarkov solution in general cases. Most recently, Yamanishi (2023) offers five ways to compute what he calls the parametric complexity – the normalization factor in the asymptotic expression in Rissanen (1996) which is essentially the same as what we have called the NML or NMPLE. Of great utility, Yamanishi (2023) Chap. 1.5 provides a discussion of the relative strengths of the computational approaches.

The Shtarkov solution in its Bayes or frequentist form comes up frequently because it defines a code length that can be used for many statistical purposes including model selection, see Hirai and Yamanishi (2013) and Suzuki and Yamanishi (2018), as well as prediction, see Kotłowski and Grünwald (2011) amongst others.

4. A computer science view of data streams

Computer science has developed its own terminology and techniques for streaming data. The algorithms are designed for rapidly arriving, massive data that cannot be stored as is. Hence, both running time and storage bounds have to be met. Here is one representative example. It is estimated that there are several hundred million routers in the world at this time of writing. Each sends and receives IP packets; contemporary routers handle a few million IP packets per second. This amounts to over a terabyte of data per day per router. To be useful, statistical analysis of the traffic through routers has to be done in real time or nearly real time and scale to the volume of data.

The generic setting for analyzing streaming data presumes discrete data arriving at discrete time points, that, as far as possible, should only be looked at once. Each data point – called a 'token' – is assumed to be in a set of the form $[U] = \{1, \dots, U\}$ for some $U \in \mathbb{N}$ where U is very large; U is sometimes called the universe. Thus, following standard computer science practice, we have a stream $\sigma = (y_1, \dots, y_n, \dots)$ simply regarded as a string of tokens without any stochastic properties. The overall goal is to extract whatever information we can from σ by identifying whatever regularities it has. As a practical point, we would also like to enforce a storage bound i.e., an upper bound on random access working memory, of the form $\ln n + \ln U$ bits. Note that no relationship between the sizes of U and n can be assumed.

Despite regarding the data as non-stochastic, computer scientists often use probability modeling to design procedures for analyzing data streams. Sometimes probability is used to analyze the performance of a method, i.e., a method is strictly empirical and probabilities are used to assess its performance under various scenarios. This is much like what the statistician faces in fixed effects linear regression: The least squares criterion can be used to define an estimator for the regression parameters but its performance depends on the error structure assumed. For instance, assuming independent and identical measurement errors ϵ_i's on the y_i's gives different inferences from assuming a serial dependence structure. In these cases, computer scientists, like statisticians, are thinking of $\mathcal{M}$-open data and considering how to impose probabilistic assumptions that make the problem $\mathcal{M}$-complete or -open. We will return to this sort of empiricism in Sec. 5.

Instead of assuming a probability structure on the data, probability is used to form the predictor or other object used for inference. That is, the computer scientist may randomly choose elements, such as hash functions defined below, from a set of elements and use the selection to form the object that will be used for inference. The techniques here mimic techniques from data compression in information theory that use random coding algorithms, usually to achieve some sort of data compression. This latter form of modeling – intended for $\mathcal{M}$-open problems – is the focus of this section. The material we present here is derived substantially from the expositions Chakrabarti (2025), Muthukrishnan (2005), Muthukrishnan (2009), and Chakrabarti et al. (2024) amongst others.

While no treatment of the topic can be exhaustive, many commonly occurring problems in streaming data have been well-studied in computer science leading to a plethora of procedures. These problems have good solutions i.e., achieve satisfactory bounds on storage and running times, even if they are still open to development. Here are five amongst many:

1. Estimating the number of distinct elements in σ;
2. Estimating the probabilities of the distinct elements in σ;
3. Identifying the most commonly occurring elements in σ;
4. Estimating moments and quantiles of σ;
5. Finding a representative sample for σ.

Getting useful answers to the first two problems is the bare minimum needed for prediction. Problem 3) is an extension of problem 2). Problem 4) is a way to find predictors such as a moments or median values and a good solution to problem 5) is the natural way to permit the streaming usage of any pre-specified predictor: simply evaluate it on the representative subset, update the subset as more data is received, and predict again using the updated representative subset.

In this section, statistical terms like estimate, moments, and quantiles are understood to be in a streaming sense, not in the sense of having a fixed finite dimensional sample space. For instance, a streaming median is not defined in the usual way – any λ satisfying $P(Y < \lambda) = P(Y > \lambda)$ or by some sort of stochastic process generalization of this – because it is allowed to evolve with the stream; see Chakrabarti et al. (2024), Sec. 11, for an alternative treatment from a computer science perspective. Often, the computer science approach to streaming data has a Bayesian feel because, as will be seen, it is standard to put a probabilistic structure not on the data but on components of the objects used for inference. This is not Bayesian in any orthodox sense but is analogous to using a prior in hierarchical Bayes.

The rest of this section proceeds as follows. First, we give some of the formalities needed for the computer science treatment of streaming data. After that, we explain one well-known procedure, the $\mathsf{Count} - \mathsf{Min}$ sketch, and give a numerical example since the thinking behind it will be unfamiliar to most statisticians. The output of the $\mathsf{Count} - \mathsf{Min}$ sketch generates a DF that can be used for prediction. Finally, we develop a statistical view of the $\mathsf{Count} - \mathsf{Min}$ sketch to see that it meshes well with established statistical treatments.

4.1. Some formalities

There are four important pre-requisite concepts to introduce.

First, we need the concept of a data structure: this is a formal way to organize, store, and present the data we have accumulated so we can compute with it. Some of this amounts to making the data 'analysis ready', but this also includes ensuring the way the data are presented can be readily updated sequentially to generate a desired output.

Second, the typical algorithm for streaming data has a well defined data structure that goes through three steps. It begins with an initialization done before beginning to process σ, followed by a processing stage that iterates with each y_i, i.e., each time a token is received. Finally, there is an output that answers our question about the stream, whether it is a prediction, a decision, or response to some other 'query'. These algorithms are called streaming if they iteratively process a data stream σ to provide their output in 'one pass' subject to a memory and running time constraint. Here, one pass means that the algorithm looks at each token in sequence once and never goes back. It updates its output from the data structure from time step to time step by summarizing the received data in a way that prevents the summary from growing too fast in size. There are multi-pass algorithms but they are less desirable.

Third, we need the concept of a sketch. Roughly, in computer science, a sketch is an algorithm that compresses the data stream so that some function of the stream can be effectively computed. More formally, a sketch is a data structure, say DS, that can be regarded as a function from each $\sigma_n = (y_1, \ldots, y_n)$ to an output at time n, say y_n^*, i.e., $DS : \sigma_n \to y_n^*$, with a 'concatenation' property. That is, if the concatenation of two streams σ and τ is denoted $\sigma \circ \tau$, there is a space efficient algorithm, say COMB that can combine them so that

$$\mathsf{COMB}\left((DS(\sigma), DS(\tau)\right) = DS(\sigma \circ \tau).$$

That is, there is an efficient way to derive the sketch for the concatenated stream from the individual streams. Essentially, the DS organizes the data in some way that lets the output of two streams be merged easily. There are many important sketches described in the references; more recently see, for example, Bahri et al. (2018) who developed sketches for classification in a Bayesian context.

Fourth, probabilistic hash functions are, arguably, the key mathematical quantities that make one pass sketches with storage and running time bounds effective for statistical tasks with streaming data. By definition, a hash function is any function that assigns fixed length values to its argument. In practice, a hash function typically maps a finite set of objects onto a smaller set of objects. As a result, hash functions are not one-to-one. On the other hand, they provide data compression: the lack of one-to-one-ness means that information in the range is strictly less than the information in the domain. The idea is that the information loss is not very big or not very important. Moreover, typically, sketches use many probabilistically chosen hash functions and this controls the information loss. The effectiveness of this form of data compression requires n be very high compared to other inputs to the sketch such as U, V, the number of distinct objects in the stream (which must also be large), etc. These are the scenarios in which the benefits of data compression outweigh the cost of using multiple hash functions.

While there are good sketches for all five tasks listed in the first part of this section, we only exemplify them here by one technique for obtaining a streaming DF in the continuous case.

To begin, fix sets $[U]$ and $[V]$ with $U > V$ and let

$$\mathcal{H} \subseteq \{h : [U] \to [V]\}.$$

The class $\mathcal{H}$ is called a hash family and the elements of $\mathcal{H}$ are the hash functions. Clearly $\#(\mathcal{H}) \leq U^V$ and it would take at most $V \ln U$ bits to encode all of $\mathcal{H}$. Let W be a probability distribution on $\mathcal{H}$ with the property that $\forall u, u'$ $\forall v, v'$: $u \neq u'$ implies

$$W(\{H(u) = v\} \cap \{H(u') = v'\}) = \frac{1}{V^2}, \tag{61}$$

where H is the random variable with outcomes $h \in \mathcal{H}$. Formally, expression (61) is called (strongly) 2-universal because two conditions on H are imposed[3]. Points at which a hash function h is not one-to-one give 'collisions': a collision occurs for h when there are $u, u' \in [U]$ so that $h(u) = h(u')$. Under (61), this happens with probability $W(H(u) = H(u')) = 1/V$ and controls the number of times we lose information about a value in $[U]$ by assigning the same v to two different u's. The smaller V is, the more collisions will occur but the greater the data compression will be.

The point of using the functions in $\mathcal{H}$ is to give up one-to-oneness so the functions can be coded with less storage by allowing for a smaller V. To get around the resulting collisions, the sketches will use multiple hash functions and combine them. It is understood (but not examined here) that these hash-function based techniques are used in settings where the storage for multiple hash functions will be less than the storage for the single correct function.

[3] Strictly speaking, 2-universal only requires '$\leq 1/V^2$' in (61). However, here we are assuming that $\mathcal{H}$ contains all possible h's so we get equality.

Here, we only address some aspects of the second problem on the list at the start of this section, namely, estimating probabilities of distinct elements in a stream, for the sake of illustrating the main ideas. Essentially, we give a one pass sketch based on probabilistically chosen hash functions. The surprising point is we can extend this sketch to continuous data, obtain an estimated empirical distribution function, and use it to predict the next outcome.

4.2. Mechanics of the Count-Min *Sketch*

In its most basic form, the Count-Min sketch is an effective way to estimate the frequencies of events in a stream when the number of distinct events is very large. Normalizing the estimated frequencies gives estimates of 'probabilities' without having to give a precise definition of probability in $\mathcal{M}$-open contexts. The normalized estimated frequencies can be used to form an *estimate* of the empirical DF (EDF), when the actual EDF would be ineffective to use, for instance when the range of the data is too large. Count-Min sketches are actually a class of procedures that give a provably good approximation to exact counts or frequencies of values under a storage bound.

Aside from being practical in some settings, using the estimated EDF (EEDF) is reasonable from a principled standpoint. In $\mathcal{M}$-open problems, there is no defined population, let alone a DF. Indeed, for $\mathcal{M}$-open problems, neither the EDF nor the EEDF need converge to a limit so by using the EEDF we are tracking the EDF along the stream.

To see how this sketch makes this possible, it is worth going through its details and then giving a numerical example. The reason is that the use of sketches and data structures (together) is quite different from how statisticians use probability modeling for data even though the two approaches play analogous roles.

First, consider the naive approach of simply constructing an EDF. Start with a data stream $\sigma = (y_1, y_2, \ldots)$ assuming values in a finite set $[U]$, see Cormode and Muthukrishnan (2005). Let $a(i) = (a_1(i), \ldots, a_U(i))$ be the number of occurrences of each $u = 1, \ldots, U$ up to time n. That is, for each $u \in [U]$ let

$$a_u(n) = \mathbf{card}\left(\{y_i |\ i \leq n \text{ and } y_i = u\}\right).$$

We update the vector $a(n)$ to $a(n+1)$ upon receipt of y_{n+1} by incrementing its u-th coordinate by one. That is, for each u

$$a_u(n+1) = \begin{cases} a_u(n) + 1 & \text{if } y_{n+1} = u \\ a_u(n) & \text{if } y_{n+1} \neq u. \end{cases} \tag{62}$$

Obviously, $a(n)/n$ is a probability vector on $[U]$ at time n. So, $a(n)/n$ gives an EDF $\hat{F}_n$ on $[U]$.

Even though $\hat{F}_n$ is not an EDF or estimated DF for any random variable, it can be used to generate a point predictor for y_{n+1}: simply find the mean, median, or other location estimator it defines and choose the value of $[U]$ closest to it. This can be extended to continuous streams and to give PI's.

There are at least three problems with this approach if U is very large. First, it can be inefficient if we insist on only using one pass procedures. Second, we want to control the storage. Third, we want to control the error. By using the data compression properties of hash functions we can resolve all three problems.

To begin the description of the Count-Min sketch, we assume U is given and that $\epsilon > 0$ and $\delta > 0$ have been chosen; in Subsec. 4.3 they will be used to characterize bounds on the error of the procedure. For the moment, we set $V = \lceil 2/\epsilon \rceil$ and choose $d = \lceil \ln(1/\delta) \rceil$ hash functions $h_1, \ldots, h_d$ independently at random from $\mathcal{H} = \{h : [U] \rightarrow [V]\}$ using a probability that satisfies (61).

Given these choices, we define a data structure for the Count part of the Count-Min sketch. For each i, we form a $d \times V$ matrix

$$C(i) = ((c_{jv}(i)))_{j=1,\ldots,d;v=1,\ldots,V}$$

that is initialized at zero for $i = 0$. For $i \geq 1$, we update $C(i-1)$ by setting each $c_{jv}(i) = \mathsf{Count}(j, h_j(y_i))$. The function Count updates here as a_u does in (62) – but using V in place of U thereby allowing collisions. That is, the (j, v) element of $C(i)$ is the number of times the j-th hash function has assumed the value $v \in [V]$ on the elements of the sequence $y_1, \ldots, y_i$. Now we have a sequence of matrices $C(0)$, $C(1)$, and so on.

We sequentially apply the Min part of the Count-Min sketch to the $C(i)$'s. Specifically, given a count matrix $C(i)$, we choose the minimum entry over the d elements in each of the v columns. That is, for each i and v we find

$$m_v(i) = \min C(i)[v] = \min_{j=1}^{d} c_{jv}(i) = \min_{j=1}^{d} \mathsf{Count}(j, v), \tag{63}$$

the minimum of the v-th column $C(i)[v]$ of $C(i)$ where v ranges over the values of $h_j(y_i)$. If we normalize this by writing $f_v(i) = m_v(i)/i$ we get a discrete probability on the elements of σ. When we use this sort of relative frequency in Subsec. 4.3, we write the corresponding EEDF simply as $\hat{F}$.

The output of the Count-Min sketch is the estimated frequencies of the tokens in σ and these values have the nice properties we want. Specifically, they are readily computed iteratively i.e., in one pass, they are combinable in the sense that if we have two streams we can add their two data structures as matrices. Less obviously, we can also control the error and the storage required. Separately, the EEDF from the estimated frequencies has nice statistical properties if we treat the problem as $\mathcal{M}$-closed or $\mathcal{M}$-complete. That is, if we compare the EEDF $\hat{F}$ (that we have) to the EDF $\hat{F}_n$ (that we only have conceptually) we can show it has the usual convergence properties we would expect.

To get a sense for what how this class of sketches is actually computed, consider a toy example. Let σ be stream formed from four values $\{A, B, C, D\}$ and suppose the first 10 elements are $[A, B, C, A, A, C, D, B, D, A]$. Choose $\delta = e^{-5}$ and $\epsilon = 2/3$. Then, $d = 5$ and $V = 3$ so we want five randomly chosen pairwise independent hash functions each taking $\{A, B, C, D\}$ to $\{1, 2, 3\}$. Suppose these are given by the columns in the table:

	h_1	h_2	h_3	h_4	h_5
A	1	1	1	1	3
B	2	2	1	3	2
C	3	2	2	2	1
D	3	3	2	2	1

The data structure of the Count-Min sketch is a 5×3 array where the j^{th} row corresponds to the j^{th} hash function and the columns correspond to the three possible values a hash function can assign to the elements of the stream. Initially all entries are 0. For the Count part, each element of the stream is passed through all of the hash functions. When an item y_i appears in the stream the count of the cell corresponding to $(j, h_j(y_i))$ increases by 1 for each j. Starting with $y_1 = A$, $h_1(A) = 1$. So the count of the cell (h_1,1) increases and the frequency changes from 0 to 1. Since $h_2(A) = 1$ also, the count of the cell (h_2,1) increases by one and its frequency changes from 0 to 1. Again, $h_3(A) = 1$. So the frequency of (h_3,1) changes from 0 to 1. Likewise, $h_4(A) = 1$. So, the cell corresponding to (h_4,1) changes from 0 to 1. Finally for this iteration, $h_5(A) = 3$ and hence the cell frequency of (h_5,3) gets updated from 0 to 1. So the 5×3 table of zeros is updated to the following table.

	1	2	3
h_1	~~0~~ 1	0	0
h_2	~~0~~ 1	0	0
h_3	~~0~~ 1	0	0
h_4	~~0~~ 1	0	0
h_5	0	0	~~0~~ 1

The second element of the stream is $y_2 = B$. Like the first element A we shall pass this element of the stream through all the hash functions. We have, $h_1(B) = 2, h_2(B) = 2, h_3(B) = 1, h_4(B) = 3, h_5(B) = 2$. So, the frequencies of the cells corresponding to $(h_1, 2)$, $(h_2, 2)$, $(h_3, 1)$, $(h_4, 3)$, and $(h_5, 2)$ are incremented by one. This is the second time we are incrementing the cell corresponding to $(h_3, 1)$ so it moves from 1 to 2. The newly updated table is the following.

	1	2	3
h_1	1	~~0~~ 1	0
h_2	1	~~0~~ 1	0
h_3	~~1~~ 2	0	0
h_4	1	0	~~0~~ 1
h_5	0	~~0~~ 1	1

We use the same procedure for the remaining 8 elements of the stream. The sum across rows, i.e., evaluations of each hash function, is 10, the length of the stream. The final table coming out of Count is the following.

	1	2	3
h_1	4	2	4
h_2	4	4	2
h_3	6	4	0
h_4	4	4	2
h_5	4	2	4

Next, we apply the Min stage. That is, we obtain an estimate for the frequency of each element in the stream from the last table. In this stage, we take the minimum of the counts over j i.e., corresponding to the cells $(j, h_j(y_i))$ as j ranges over $1, 2, \cdots, d$. Thus, for $n = 10$, we see from the final table that for A, $\mathsf{Count}(h_1, 1) = 4$, $\mathsf{Count}(h_2, 1) = 4$, $\mathsf{Count}(h_3, 1) = 6$, $\mathsf{Count}(h_4, 1) = 4$, and $\mathsf{Count}(h_5, 3) = 4$. (to be clear, we see that, in the last value for instance, that the '4' is from cell (5,3) because $h_5(A) = 3$.) Now, the estimated frequency of A is $\min(4, 4, 6, 4, 4) = 4$.

Similarly, for B, we have $\mathsf{Count}(h_1, 2) = 2$, $\mathsf{Count}(h_2, 2) = 4$, $\mathsf{Count}(h_3, 1) = 6$, $\mathsf{Count}(h_4, 3) = 2$, and $\mathsf{Count}(h_5, 2) = 2$. So the estimated frequency of B is $\min(2, 4, 6, 2, 2) = 2$. Doing the same for C and D gives the output of the $\mathsf{Count} - \mathsf{Min}$ sketch. Here is the comparison of the actual frequencies and the estimated frequencies for the distinct elements in the finite stream:

σ	Actual Frequency	Estimated Frequency
A	4	4
B	2	2
C	2	4
D	2	2

In this table, we got exact equality for for A, B, and D but the value for C is a little bit higher. In general, the estimated frequency for each element is greater than or equal to the actual frequency because of collisions in the hash functions, i.e., when $h_j(y) = h_j(y')$ for some $y \neq y'$. For streams with many more possible elements the agreement will typically not be nearly as good unless n is very large. Obviously, in practice we would go through the Count part followed by the Min part for each i and use the resulting probability to issue predictions or other decisions for the next time step.

4.3. Statistical aspects of the Count-Min *Sketch*

Recall the setting of Subsec. 4.2, i.e., we have a stream $\sigma = (y_1, y_2, \ldots)$ from $[U]$ and are given $\epsilon, \delta > 0$. So, we have $d = \lceil \ln(1/\delta) \rceil$ randomly chosen 2-universal hash functions $h : [U] \to [V]$ where $V = \lceil 2/\epsilon \rceil$ leading to an EEDF $\hat{F}$ for the EDF $\hat{F}_n$ for a discrete stream. We start by extending from a discrete stream to a continuous stream and then state some properties of the extension. These properties will be from an $\mathcal{M}$-closed or -complete standpoint because otherwise it is unclear how to give formal properties. This means we are assuming the methods are good for $\mathcal{M}$-open cases because they perform as they should in $\mathcal{M}$-closed and -complete settings.

4.3.1. Extension to continuous streams

To discuss the statistical properties of the output from the $\mathsf{Count}-\mathsf{Min}$ sketch, we convert from discrete outcomes to streams of continuous outcomes. It is straightforward to do this and continuous data is convenient for many analytic purposes, e.g., taking limits is easier than trying to relate specific rates of increasing $[U]$, $[V]$, and d to the behavior of $\hat{F}$.

Consider a stream $(y_1, y_2, \dots, y_n, \dots)$ with $y_i \in \mathbb{R}$ for $i = 1, \dots, n$. Fix $M > 0$, $K \in \mathbb{N}$, and partition $[-M, M]$ into K intervals each of length $2M/K$. In fact, K will play the role of U but be under our control. Denote each interval by

$$I_{\mathsf{k}} = I_{\mathsf{Kk}} = \left(-M + (k-1)\frac{2M}{K}, -M + k\frac{2M}{K}\right]. \tag{64}$$

Also, let $I_{\mathsf{0}} = (-\infty, -M)$ and $I_{\mathsf{K}+1} = (M, \infty)$. In practice, if the stream y^n is bounded e.g., $M_1 \le y_i \le M_2$, it is convenient to modify these definitions so the intervals only cover $[M_1, M_2]$. Indeed, in our computations, we take $M_1 = 0$, fix an upper bound M_2 and can ignore I_0 and I_{K+1}. Our goal will be to use the $\mathsf{Count}-\mathsf{Min}$ sketch on the discretization of $\mathbb{R}$ to produce an EEDF that we can use to predict y_{n+1} after seeing $(y_1, \cdots, y_n)$.

To link the y_i's to the I_k's, let

$$a_{\mathsf{k}} = a_{\mathsf{Kk}}(n) = \#\{y_i \in I_{\mathsf{k}} \mid i = 1, \cdots, n\}$$

so that $a_k(n)$ is the frequency of the tokens in $(y_1, \cdots, y_n)$ that fall in I_{k}.

Let $d = d_{\mathsf{K}}$ and randomly choose hash functions $h_1, \cdots, h_{d_{\mathsf{K}}}$ where, for $j = 1, \dots, d_{\mathsf{K}}$, each $d_j : [K] \to [W_K]$ for some W_{K} that plays the role of V. For storage bounds, we want $W_{\mathsf{k}} \ll \mathsf{K}$. More generally, we want W_{K} and d_{K} small as functions of K but we will have to allow them to increase slowly with K while K itself increases.

We extend the discrete h_j's to $\mathbb{R}$ by defining

$$\tilde{h}_j : \mathbb{R} \longrightarrow \{0, 1, 2, \cdots, W_{\mathsf{K}}, W_{\mathsf{K+1}}\}$$

where $\tilde{h}_j(s) = h_j(\mathsf{k})$ for $s \in I_{\mathsf{k}}$ and $\mathsf{k} = 0, 1, \dots, \mathsf{K}, \mathsf{K}+1$. In terms of the tokens, this means for $i \le n$ and $y_i \in I_{\mathsf{k}}$ we have

$$\tilde{h}_j(y_i) = h_j(k).$$

Following the $\mathsf{Count}-\mathsf{Min}$ procedure as described in Chanda et al. (2026), we define an estimate of a_{k} (frequency of the k^{th} interval) at time n. For the j^{th} hash function h_j, an interval k and time n, we set

$$\hat{a}_{jk} = \mathsf{Count}_n(j, h_j(k)) = \#\left(\{i \le n \mid \tilde{h}_j(y_i) = h_j(k)\}\right)$$

so that the estimate $\hat{a}_k$ of a_k becomes

$$\hat{a}_k(n) = \min_j \ \hat{a}_{jk}(n) \ge 0. \tag{65}$$

Now, the estimated EDF (EEDF) generated by the $\mathsf{Count} - \mathsf{Min}$ sketch is

$$\hat{F}(x) = \sum_{k \leq x} \frac{\hat{a}_k(n)}{n} \tag{66}$$

and the actual EDF is

$$\hat{F}_n(x) = \sum_{k \leq x} \frac{a_k(n)}{n}. \tag{67}$$

The EEDF is only an estimate of the EDF because the $\mathsf{Count} - \mathsf{Min}$ sketch only gives an estimate of the frequencies. The reason is that the EEDF is based on hash functions so that it will satisfy a storage bound that we will shortly state.

To finish the present line of reasoning, we use (66) to define point predictions. (The EEDF also gives interval predictions, but it is unclear what the interval means in $\mathcal{M}$-open settings.) In our computed results we use two predictors:

1. Weighted mean: our prediction is the weighted mean of the midpoints of the intervals $I_\mathsf{k}; \mathsf{k} = 1, 2, \cdots, \mathsf{K}$ defined in (64) for some large K and M, where the weights are $\hat{a}_\mathsf{k}$ as defined in (65). Formally, denote the mid-point of interval I_k by m_k. Then,

$$\hat{y}_{n+1} = \hat{y}_{\mathsf{K},n+1} = \sum_{\mathsf{k}=1}^{\mathsf{K}} m_\mathsf{k} \frac{\hat{a}_\mathsf{k}(n)}{n} \tag{68}$$

2. Weighted median: our prediction is the weighted median of the m_k's, with weights $w_\mathsf{k} = \hat{a}_\mathsf{k} / \sum_{\mathsf{k}=1}^{\mathsf{K}} \hat{a}_\mathsf{k}$, defined as the average of m_{q-1} and m_q, where m_q satisfies

$$\sum_{i=1}^{q-1} w_i \leq \frac{1}{2} \text{ and } \sum_{i=q+1}^{K} w_i \leq \frac{1}{2}. \tag{69}$$

4.3.2. Desirable properties of the extension

Here we state four results for use with streaming $\mathcal{M}$-open data: an error bound, a storage bound for the error bound, a convergence result for $\hat{F}$ to the EEDF, and a convergence result for $\hat{F}$ to F, when it exists. The mode of convergence to the EEDF is defined by the probability on the hash functions not by any probability associated with the stream, cf. the discussion in Chanda et al. (2026). The mode of convergence of $\hat{F}$ to F includes the probability associated with F.

Our first result is that $\hat{a}_\mathsf{k}$ is a good estimate of the frequency of an interval I_k, essentially a consistency result for fixed K. Let $||a||_1 = \sum_{\mathsf{k}=1}^{\mathsf{K}} a_\mathsf{k}(n)$ be the sum over k of the number of elements in y up to time n that land in I_K, where K and n are suppressed in the notation $||a||_1$. We have the following statements, similar to the guarantee for the $\mathsf{Count} - \mathsf{Min}$ sketch; see Muthukrishnan (2005), Muthukrishnan (2009), and Chanda et al. (2026).

Proposition 4.1. *Let W correspond to the probability in* (61)*. Then, $\forall \epsilon > 0$ and $\forall \delta > 0$, $\exists\ N$ such that $\forall d_{\mathsf{K}} > N$, we have*

$$W(\forall j = 1, \cdots, \mathsf{d}_{\mathsf{K}}; \hat{a}_{jk}(n) \leq a_{\mathsf{k}}(n) + \epsilon ||a||_1) \leq \delta.$$

Remark. Here, $\|a\|_1 = n$ because we are looking at data streams in the cash register model of streaming data i.e., items only accumulate. It is immediate from Prop. 4.1 that we get $\hat{a}_{\mathsf{k}}(n) \leq a_{\mathsf{k}}(n) + \epsilon ||a||_1$ from the minimum in (65). Also, by construction we get $a_{\mathsf{k}}(n) \leq \hat{a}_{\mathsf{k}}(n)$. So, we have upper and lower bounds.

Next, we address the storage requirement for the procedure used in Prop. 4.1. Heuristically, observe that the storage is upper bounded by the number of hash functions $\ln(1/\delta)$ multiplied by the number of values each hash function can take, namely e/ϵ giving $\mathcal{O}((1/\epsilon)\ln(1/\delta))$. Adapting the proof in Muthukrishnan (2009) to our present setting, we see that storage of the order $\mathcal{O}(1/\epsilon)$ will suffice, see Chanda et al. (2026).

Theorem 4.1. *Let $\eta > 0$. Assume the storage available is $\Omega(1/\epsilon)$[4]. Then, under the probability in* (61)*, we have that*

$$P(\hat{a}_{jk} \leq a_k + \epsilon ||a||_1) \leq \eta.$$

We extend Prop. 4.1 by letting $\mathsf{K}, \mathsf{d}_{\mathsf{K}}, n \to \infty$ at appropriate rates to get a consistency result for the EEDF. That is, our EEDF converges to an EDF based on the streaming data that is not necessarily the true DF since it needn't exist. We have the following.

Theorem 4.2. *Let $x \in (0, M]$. Then, pointwise in x,*

$$\hat{F}(x) - \hat{F}_n(x) \overset{W}{\to} 0 \text{ as } \mathsf{d}_{\mathsf{K}}, \mathsf{K}, \text{and } n \longrightarrow \infty.$$

Unsurprisingly, if the stream comes from an $\mathcal{M}$-closed or -complete source F, then the EDF converges to F (by the usual law of large numbers) and so does the EEDF $\hat{F}$. We state this as the following.

Corollary 4.1. *If there exists an F such that the Y_i's are independently and identically distributed according to F, then, under the conditions of Theorem 4.2, we have that the $\mathsf{Count} - \mathsf{Min}$ sketch generated estimate $\hat{F}$ of F is consistent for F, that is*

$$\hat{F} \longrightarrow F,$$

in the joint mode of convergence defined by the W used in Theorem 4.2 and the DF the Y_i's follow.

Like the EDF, the EEDF tracks the location of the data. Recall that a $100(1-\alpha)\%$ PI is given by $(\hat{F}_n^{-1}(\alpha/2), \hat{F}_n^{-1}(1-\alpha/2))$. As n increases, the EEDF follows the data i.e., the relationship between y_n and y_{n+1} may be different from the relationship between y_{100n} and y_{100n+1}. The location of $\hat{F}$ moves to track where

[4] Ω-notation gives a lower bound in contrast to big-$\mathcal{O}$ notation that gives an upper bound.

the preponderance of data is. There are relatively standard methods, see Cesa-Bianchi and Lugosi (2006), to force more recent data to be weighted more and these techniques can be combined with the EEDF if desired.

To conclude this section, we make a few observations about how $\hat{F}$ can be expected to behave. First, it is possible to prove a Glivenko-Cantelli Theorem for the convergence of $\hat{F}$ to $\hat{F}_n$ and again observe the reduction when the Y_i's follow an F. We cite Chung (1974) for the standard form and proof of such theorems; see also Shaikh (2022); cf. Chanda et al. (2026). We want results about $\hat{F}$ because it is fully empirical. These results can likely be generalized to many dependent data settings. Likewise, we suggest versions of other major theorems for the EDF such as Donsker's theorem and the Kiefer-Dvoretzky-Wolfowitz theorem can also be established for $\hat{F}$.

5. Conformal prediction

The central premise of conformal prediction (CP) is that future data looks like past data, i.e., they 'conform'. Leaving aside the philosophical question as to what that means in $\mathcal{M}$-Open settings, the practical implication is that we find data dependent quantities and associate a prediction interval to them. That is, we treat the data received as a set of real numbers and form expressions for a future value without reference to any probability structure.

A caricature of conformal methods would be the following. Take the mean and SD of n observations and announce $\bar{y} \pm c_\alpha \hat{\sigma}$ as a $1 - \alpha 100\%$ PI provided we had some way to interpret the α e.g., perhaps using a percentile from an EDF based on the accumulated data, but not necessarily saying that the data followed any distribution. Then, make distributional assumptions about the data and investigate the behavior of the predictor under those assumptions. See Lei et al. (2018) for an analysis of conformal predictors under stochastic assumptions.

This idea is far from new: in standard linear regression we obtain estimators as a result of an optimization under squared error and only derive inferential properties of them after specifying an error structure. Indeed, in the conventional setting, the metric structure (squared error) is logically independent of the error structure (normality) and there are multiple viable choices for both.

It is easy to see that conformal prediction is applicable to $\mathcal{M}$-open problems because, like many computer science techniques, it is purely empirical and does not necessitate assumptions to be stated explicitly. We only use the error structure to prove distributions properties.

Accordingly, in this short section we only present the empirical aspects of conformal prediction, ignoring the accumulated analyses of these methods in $\mathcal{M}$-complete and -closed settings. Our discussion owes much to the 'computer science view' as expressed in Shafer and Vovk (2008), Balasubramanian et al. (2014), and Vovk et al. (2022).

5.1. No explanatory variables

CP is based on the concept of a conformity measure. There are many choices, but each of them provides an analog of concept of confidence for PI's. This is empirical and does not invoke an error structure. A nonconformity measure assesses how close a potential future value y_{n+1} is to the accumulated data $y_1, \ldots, y_n$. Formally, conformity measures are non-negative functions $C : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$ assumed symmetric in their first argument. They formalize the idea of distance between a set of outcomes of size n and a future outcome. It is tempting to think of these sets as y^n and y_{n+1} respectively, but the sense of conformity used here is more subtle and has a feel of cross-validation.

To understand the intuition, fix a data vector y^{n+1} where y^n is the data we have and y_{n+1} is a candidate future value. Let $i \in \{1, \ldots, n+1\}$ and remove y_i from y^{n+1}. Write the y_i-deleted y^{n+1} as $y^{n+1\backslash i} = (y_1, \ldots, \hat{y}_i, \ldots, y_{n+1})$, where the hat indicates the deletion.

We use C to compare y_i with $\hat{y}^{n+1\backslash i}$. So, write

$$C_i(y_{n+1}) = C(\hat{y}^{n+1\backslash i}, y_i).$$

Given that we have chosen C properly, $C_i(y_{n+1})$ is large when $\hat{y}^{n+1\backslash i}$ 'conforms' with y_i (because of y_{n+1}), intuitively when y_i is close from the 'middle' of the values in $y^{n+1\backslash i}$. Now, the proportion of times, out of $n+1$, that $C_i(y_{n+1}) \leq C_{n+1}(y_{n+1})$ measures how similar y_{n+1} is to y^n. We use $n+1$ because for $i = n+1$ the inequality holds trivially – $C_{n+1}(y_{n+1}) \leq C_{n+1}(y_{n+1})$ by definition. The proportion of times the reverse inequality holds measures how similar y_{n+1} is to y^n. Thus, if this proportion is high enough we want to put the candidate value y_{n+1} into our PI. The reasoning is that the higher this proportion is, the more y_{n+1} conforms to y^n, as a set.

Let $\alpha > 0$ be the level of a conformal PI for y^{n+1} based on y^n. We define

$$PI(\alpha, y^n) = \left\{ y_{n+1} \mid \frac{\#\left(\{i | C_i(y_{n+1}) \geq C_{n+1}(y_{n+1})\}\right)}{n+1} \geq \alpha \right\}. \tag{70}$$

That is, if a value y_{n+1} gives a high enough proportion of large enough C_i's, it is put in $PI(\alpha, y^n)$. Clearly, for $\alpha_1 < \alpha_2$ we have $PI(\alpha_2, y^n) \subset PI(\alpha_1, y^n)$.

It remains to propose suitable conformity measures or, equivalently, nonconformity measures. Perhaps the easiest is the 'distance to average' (DTA) nonconformity: $C(y^n, y_{n+1}) = |\bar{y} - y_{n+1}|$. The closer the potential future value y_{n+1} is to the mean, the more it conforms to the existing data. While intuitively reasonable, this choice of C is has a computational limitation: it requires computing $n+1$ means at stage n. To avoid this, it is common to use $C(y^n, y_{n+1}) = |\bar{y}_{n+1} - y_{n+1}|$. We see that

$$C(y^n, y_{n+1}) = |\frac{n\bar{y} + y_{n+1}}{n+1} - y_{n+1}| = \frac{n}{n+1}|\bar{y} - y_{n+1}|$$

so the two forms of the DTA are equivalent as n increases.

As a generality, we can use density estimators as nonconformities. For instance, Lei et al. (2013) proposes that we set $C(y^n, y_{n+1}) = \hat{p}(y_{n+1}; y^n)$ where $\hat{p}$ is a density estimator for the true density of the stream (which may not exist) using y^n. Establishing results that ensure consistency of $\hat{p}$ identifies regularity conditions on C ensuring it reduces properly in an $\mathcal{M}$-closed problem. Focusing on kernel methods, Lei et al. (2013) give results on bandwidth selection and studied the actual prediction regions. It is possible to get prediction regions that have as many disjoint intervals as there are modes in the true density; it is not clear how to overcome this possible problem. This line of inquiry is continued in Lei et al. (2018) which re-interprets conformity in terms of probability of mis-coverage and focuses on high-dimensional prediction problems that are often ignored.

Another relatively easy choice of conformity measure is Bayesian, namely the posterior predictive density as used in Bersson and Hoff (2024). The simplest is of course based on the normal. Suppose the Y_i's are IID $\mathcal{N}(\theta, \sigma^2)$ with a normal prior assigned to θ, $\theta \sim \mathcal{N}(\mu, \tau^2\sigma^2)$ and an inverse Gamma assigned to σ, i.e., $1/\sigma^2 \sim \mathsf{Gamma}(a/2, b/2)$ for some $a, b > 0$. Then, for given μ and τ, set $C(y^n, y_{n+1}) = p(y_{n+1}|y^n)$. This choice reflects conformity in the sense of a new data point being representative of the predictive density. In Sec. 7, we use their package for Bayesian CP; there is also a CP package for the techniques in Lei et al. (2018).

Conditions to ensure that the PI is an actual interval are in Bersson and Hoff (2024) and Hoff (2023) gives conditions to ensure the region is efficient, or Bayes optimal, in the sense that the prediction region is, asymptotically, as small as possible given the level, assuming the model is true. These amount to regularity conditions to ensure the method reduces to what is should be in $\mathcal{M}$-closed problems. For models other than what Bersson and Hoff (2024) study, it is possible that PI's are union of disjoint intervals.

5.2. *Explanatory variables present*

To see how conformal prediction extends to regression with explanatory variables consider simple linear regression. For data $\mathcal{D}_n = \{(x,y_1), \ldots, (x_n, y_n)\}$ suppose we write $y = \hat{a} + \hat{b}x$. Then, for a possible predicted value $\hat{y}_{n+1}$ at x_{n+1} write the nonconformity

$$C(\mathcal{D}_n, (x_{n+1}, y_{n+1})) = |y_{n+1} - \hat{y}_{n+1}| = |y_{n+1} - \hat{a} - \hat{b}x_{n+1}|.$$

It is seen that the nonconformity uses the predictor class as an input.

Now, the procedure from Subsec. 5.1 can be generalized directly. To wit: Given $\mathcal{D}_n$ and x_{n+1}, consider the candidate future outcome y_{n+1} and write

$$C_i(y_{n+1}) = C(\mathcal{D}_{n+1} \setminus \{(x_i, y_i)\}, \{(x_i, y_i)\}).$$

We write the analog of (70) again putting values of y_{n+1} into it for x_{n+1} when a high enough proportion of the C_i's are larger than $C_{n+1}(y_{n+1})$. The procedure is the same for any other class of predictors for Y from X.

It is seen that the conformity measure assesses the degree to which a candidate new data point conforms to the earlier data and that its conformity depends on the predictor class. Thus the nonconformity of a new value with the older values will differ if a different regression technique is used. For example, the prediction will depend on whether a linear model, an RVM, or a random forest is used. This is typical for predictive techniques but it is unclear how reasonable it is for $\mathcal{M}$-open data.

Indeed, it is an open question whether $\mathcal{M}$-open data satisfies a variance-bias tradeoff. Likely it does because one predictor can be worse than another but not in any sense that can be readily formalized. Nevertheless, it is numerically reasonable to conjecture that if we have three predictor classes, small medium and large, the medium class may be better (have a higher conformity score) than the small or the large due to bias and variance, respectively. Moreover, if there are two predictor classes leading to two different conformity measures, it is possible to find the corresponding PI's for a given level α but the ordering of the two sets of C_i's may not match and they may not be as comparable as desired. However, the performance of the PI's from two different conformities can be empirically compared in terms of coverage, for instance. These problems become more complex if two different nonconformity measures e.g., a Bayesian's posterior versus a kernel density estimate, are used as well as two classes of predictor families.

Work by Johansson et al. (2014) compares conformities based on random forests, neural nets, and nearest-neighbors methods and argues that random forests typically provide the most efficient prediction regions for certain fixed conformities. Recent work by Diniz et al. predictively compares Bayesian linear regression to ridge regression and argues that ridge regression with conformal prediction often works better.

5.3. $\mathcal{M}$-open caveats

It is important to stress that CP gives a set that under the given conformity measure has a prescribed level α and that this is strictly empirical. That is, there is no necessary error structure. CP is at root a deterministic way to construct an interval that reflects where the data were and by assumption where the next data point will be. It is motivated by the standard behavior of exchangeable stochastic processes. Here, for point prediction purposes, we simply average the endpoints of the interval (with $\alpha = .85$).

The good statistical properties of CP are generally in a frequentist sense; the weakest assumption typically used is that the DG is exchangeable and exchangeability is usually regarded as a minimal assumption. This may be reasonable for many modeling scenarios that use stochastic processes, but is inappropriate for $\mathcal{M}$-open problems. Indeed, here, with streaming data, we have also used a representative subset. So, our methods accommodate data that is not stationary. That is, the DG may 'drift' and this will be detected over time by the streaming representative subset.

More specifically, our methods here, and CP in particular, allow for a DG that evolves in an unpredictable way over time. This happens for instance in some of the rainfall data, see Fig. 12 in Appendix D.1. This allowance may be in a sudden change point sense or continuous.

6. Neural networks

Neural nets (NN's) were originally proposed as a model for neurons. They were quickly ruled out as a physical model but recognized as a promising technique for nonlinear regression or classification that might be useful, especially with multitype data. Here we develop NN's for prediction with streaming data in three steps. First, we describe general fully connected feedforward NN's. Then we extend these to recurrent NN's, RNN's. RNN's then can be extended to Long Short-Term Memory NN's, LSTM's, which are the most common version used in practice. We only present the latter for the case of streaming unidimensional data, but it will be clear how they generalize to include explanatory variables and multidimensional streaming data.

6.1. Feedforward NN's

The simplest NN is a function. Specifically, if there are explanatory variables $x = (x_1, \ldots, x_p)^T$, write

$$y = \sigma(x^T\beta + \nu) \tag{71}$$

where $\beta = (\beta_1, \ldots, \beta_p)^T$ defines a linear operation, $\nu \in \mathbb{R}$ represents bias, and σ called is an activation function i.e., $\sigma : \mathbb{R} \to [0, 1]$ and satisfies $\sigma(-\infty) = 0$ and $\sigma(\infty) = 1$. The most common activation function is

$$\sigma(x) = \sigma_{\beta,\nu}(x) = \frac{1}{1 + \exp(-(x^T\beta + \nu))}.$$

Expression (71) is called a node function since it is usually represented by a circle, as in Fig. 3. Usually, (71) is 'stochasticized' into a signal plus noise model by writing

$$Y = \sigma(x^T\beta + \nu) + \epsilon, \tag{72}$$

where ϵ is a noise term, usually taken as independent over samples.

Clearly, explanatory variables can be fed into multiple node functions and the resulting node functions combined again into an overall output. This can be done repeatedly forming layers as indicated in Fig. 4. These NN's are feedforward in the sense that no output is an input to an earlier layer. These NN's are fully connected in the sense that all outputs from any layer feed into each node of the next layer. Layers strictly between the input variables (the x_j's) and the output (y) are 'hidden' because they are internal to the NN if it is regarded as a black box. If $r_2 = 0$, the result is a single hidden layer NN.

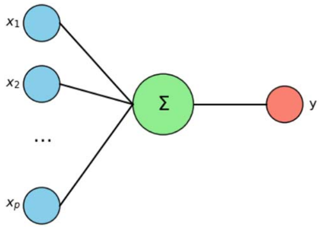

Fig 3. *The simplest NN: p variables feed into one node and the output is given by y.*

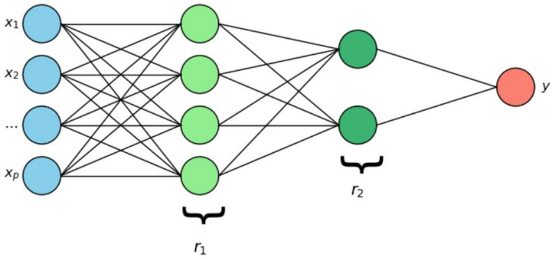

Fig 4. *A feedforward, fully connected, two hidden layer NN. The first layer has r_1 nodes; the second hidden layer has r_2 nodes.*

The equations represented by the diagram in Fig. 4 are as follows. For $k_1 = 1, \ldots, r_1$, the functions in the first hidden layer are

$$\sigma_{1,k_1} \quad = \quad \sigma(x^T \beta_{1,k_1} + \nu_{1,k_1}). \tag{73}$$

For $k_2 = 1, \ldots, r_2$ the functions in the second hidden layer are

$$\sigma_{2,k_2} = \sigma\left((\sigma_{1,1}, \ldots, \sigma_{1,r_1})^T \beta_{2,k_2} + \nu_{2,k_2}\right). \tag{74}$$

That is, the outputs of the first hidden layer are inputs to the second hidden layer and we treat them as if they were explanatory variables. The single output on the right is

$$y = \nu_3 + \sum_{k_2=1}^{r_2} \beta_{3,k_2} \sigma_{2,k_2}. \tag{75}$$

Often a noise term is added to (75). Extensions to three or more hidden layers – or two or more final outputs – are similar but notationally cumbersome.

Estimation in NN's is done in two stages. First, the network architecture is determined, i.e., the number of layers, the number of nodes per layer, and the connectivities from layer-to-layer are chosen either by a model selection technique or on the basis of modeling. The former is usually done by some stochastic model selection procedure such as simulated annealing. In the latter case, this is bypassed simply by choosing a NN that is large enough. Second, the (many) parameters are estimated usually by some version of back propagation; this is a way to use the stochasticized version of the NN to get a sum of squared errors (SSE). Essentially, the SSE is minimized incrementally as the parameter values are varied; this is usually an extension of Newton's method and can become very complicated as the number of layers increases. Often a penalty term is used in these optimizations. Originally, this was used primarily to stabilize estimates in the sense of smaller MSE's. Later, penalties were recognized as good in a shrinkage sense, that is to zero out some parameters.

In our present case of streaming data, we do not have explanatory variables so this class of NN's cannot be applied directly to y_n to generate a prediction for y_{n+1}. Obviously, one can take a collection of summary statistics from the first n data points (as we do later) and then choose an architecture, estimate the parameters, and iterate as data comes in. Aside from the evident difficulty with this, there are other classes of NN's that are considered more appropriate for streaming data. We develop these in the following.

6.2. RNN's

RNN's are generalizations of feedforward NN's. To understand what a RNN is, it is enough to construct the simplest nontrivial class. Then, generalizations to larger RNN's will only be a matter of more complex notation.

Start with a single hidden layer NN with, say r_1 hidden nodes. If we use it once, i.e., at the first time step for a single value of a p-dimensional x, we generate a single real output y. If we write the function the NN represents as f, i.e., we assume we know all the parameters, then for the first time step we get $\hat{y}_1 = f(x_1)$ and $\hat{y}_1$ is our predicted value for y_1 given x_1.

At the second time step, suppose we have x_2. To allow the 'recurrence' we take the r_1 outputs from the hidden layer at time step one and feed them into the hidden layer for time step two as if they were simply concatenated with x_2. That is, we have r_1 more nodes in the hidden layer for time step two than we had at time step one. We then repeat this: for time step three and on, we concatenate x_n with the output of the hidden layer from time step $n-1$ thereby using a single hidden layer feedforward net with $r_1 + p$ inputs to generate $\hat{y}_n$ as our prediction for y_n.

We can see this more formally by taking $r_2 = 0$ in Fig. 4 and using the resulting single hidden layer NN for time step one. That is, we generate the r_1 values in (73) and use them to produce a single output that we can denote as $\hat{y}_1 = \sigma_{2,1}$. To get the output for x_2, we concatenate x_2 and $\sigma_2 = (\sigma_{1,1}(x_1), \ldots, \sigma_{1,r_1}(x_1))$. Then we set

$$\hat{y}_2 = \gamma_2 \sigma_{2,2} = \gamma_2 \sigma(\beta^T x_2 + \nu + \beta_R^T \sigma_2) + \nu) \tag{76}$$

where γ_2 is a factor and β_R is an r_1 dimensional vector controlling how the outputs of the previous time step enter the current predictor. For step three, we update σ_2 to the corresponding value σ_3 and get $\hat{y}_3 = \gamma_3 \sigma(\beta^T x_3 + \nu + \beta_R^T \sigma_3)$ and this 'recurrence' is iterated.

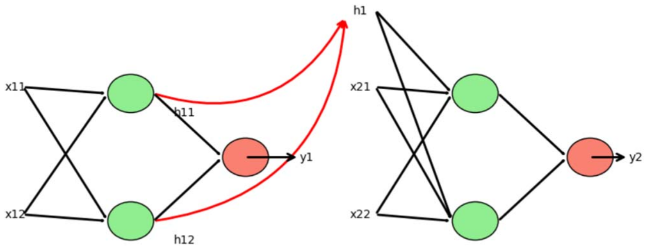

Fig 5. *This illustrates the repeated structure of a the simplest nontrivial RNN. At time step one, we feed the first values of a two dimensional explanatory variable* (x_{11}, x_{12}) *into the leftmost nodes. The output is* y_1*. At the second time step, new values* (x_{21}, x_{22}) *are fed into the network along with the output of the single hidden layer* $h_1 = (h_{11}, h_{12})$*. The pattern is repeated for time step three using a new value of the explanatory variables and the output of the hidden layer from time step two.*

This procedure is diagrammed in Fig. 5. In our present case where we have not included explanatory variables, we replace our two-dimensional x_1 with our unidimensional y_1 and initialize the network at time step one with an input $h_0 = (h_{01}, h_{02})$. Then we repeat for time step two and so on.

Conceptually, RNN's are a way to allow dependence on earlier data. They are a little like Markov chains in that the dependence is one time step back. On the other hand, the form of dependence implicitly includes earlier data. As a separate point, RNN's have computational problems. First, architecture selection is more difficult, often done by CV or simply by trial-and-error. Also, often when parameters are being estimated by 'backprop', where the 'propagation' is through time, the gradient misbehaves, going to zero or infinity. This problem worsens as n increases. As we shall see, this problem can be largely resolved by embedding RNN's into a larger network.

6.3. Long short-term memory NN's

Long Short-Term Memory (LSTM) NN's are a generalization of RNN's. They get their name because they are intended to handle both short term and long term dependencies in sequential data. It would be more accurate to call them RNN's with a cyclic memory module, but LSTM has become generally accepted. When correct and accurately estimated, the extra cyclic memory ensures that the proper amount of data from the distant past is included in current inferences. This cyclic memory is in addition to the retention of more recent data that is already ensured by the recurrence in the RNN.

Fig. 6 shows the structure of an LSTM for the context of streaming data without explanatory variables. It has five parts: an RNN; three gates: forget, input, and output; and a 'memory cell'. In computer science, 'gate' normally means a logic gate i.e., an implementation of a Boolean function. Here, the term gate is used to mean a specified mathematical function that is regarded as a module in the sense of taking relatively few inputs and generating relatively

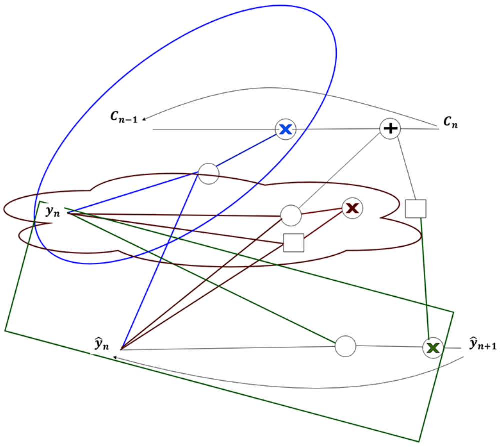

Fig 6. *Diagram of the simplest nontrivial LSTM NN that does not use explanatory variables explicitly. Circles indicate a sigmoid activation function and squares indicate a* tanh *activation function. The blue ellipse indicates the forget gate. The cloud shaped region indicates the input gate. The green rectangle indicates the output gate. The recursion along the top is the memory cell. The recursion along the bottom represents the RNN.*

few outputs. The memory cell is a module that stores data from one time step for use at the next time step. It is straightforward to extend this to include explanatory variables.

Let us go through the construction of an LSTM part-by-part.

1. Apart from the multiplication at the end, the recurrence along the bottom of Fig. 6.3 in the green box is an RNN: at the $n+1$ time step, the new y_{n+1} is combined in a node with the output $\hat{y}_{n+1}$ from the n-th time step.
2. The blue oval is called the 'forget' gate. The node with the blue times symbol multiplies the recurring C_{n+1} by the output of a node that takes the value of the explanatory variable at time $n+1$ and the recurring output $\hat{y}_{t+1}$ as inputs to the 'memory' cell. (Calling this a forget gate is also a misnomer; it really functions like an inclusion rate.)
3. The subnet that combines y_{n+1} and $\hat{y}_{n+1}$ to be input into the memory cell is called the input gate. It has two nodes. One, the circle, is a sigmoid function giving a value in $[0,1]$. It scales the output of the other, a square, that is a sf tanh function giving a number in $[-1,1]$. This is indicated by the brown times symbol. The idea is that the sigmoid controls the amount of information from tanh that will go into the memory cell.
4. The recurrence along the top line of Fig. 6.3 is called the memory cell. At time step t it transforms C_n into C_{n+1} by including the outputs of the forget gate and the input gate.
5. These are now fed into the output gate which is simply the last nodes that combine to generate $\hat{y}_{n+1}$.

If desired, the input y_n can be replaced by x_n, the values of explanatory variables at time step n. In our description here, we are simply using y_n as the explanatory variable for time step $n+1$ when y_{n+1} is observed.

Architecture selection and parameter estimation for LSTM's are more difficult than for RNN's. Nevertheless, the same general approaches are used. Usually, regularization is used in optimizing over SSE to achieve sparsity and stability. One of the arguments for LSTM's is that even though the computing is harder, it is more likely to be successful because the gradients in backprop methods tend to avoid zero and infinity.

There are many variants on this simple description of LSTM's. For instance, LSTM's are often used in levels and layers. As a generality, adding more layers (in sequence) can improve the selection of explanatory variables and adding more LSTM's within a layer increases the model's ability to 'learn', possibly at different time scales. There are many other classes of neural nets but they are beyond our present scope.

7. Computational comparisons

In this section we give computational results for 14 predictors and three data sets. Results from three more data sets are given in Appendix D. All six data sets are very complex. We regard five of them as $\mathcal{M}$-open. The sixth is, we think, $\mathcal{M}$-complete. The results for it provide a sort of counterexample: using a rich enough modeling strategy can outperform techniques designed for $\mathcal{M}$-open data when a model exists.

For each method and data set, we give cumulative predictive errors over the stream of data. We also perturb the data by adding independent $N(0, \tau^2)$ noise, for small $\tau > 0$, to each data point and recalculating the cumulative errors of each method. The resulting curves are functions of τ and assess the sensitivity of the methods.

In the next subsection, we describe our settings formally. Then we present and discuss our computational findings and make some tentative recommendations. Details for the implementations of the methods are given in Appendix D.1.

7.1. Settings for the comparison

For each of the 14 predictors, we compute the cumulative L^1 error. That is, for each method and each data stream $(y_1, y_2, y_3, \ldots)$ we have a sequence of errors $|y_{n+1} - \hat{y}_{n+1}|$ where a given prediction $\hat{y}_{n+1}$ depends on $y_1, \ldots, y_n$ (and possibly a burn-in set $\mathcal{D}_b$) and we find the cumulative predictive error

$$CPE = CPE(n+1) = \frac{1}{n+1} \sum_{i=1}^{n+1} |y_i - \hat{y}_i|. \tag{77}$$

It is seen that

$$CPE(n+1) = \frac{1}{n+1} \left(nCPE(n) + |y_{n+1} - \hat{y}_{n+1}| \right)$$

so we can easily compute it recursively. Note that this assessment follows the prequential principle; see Dawid (1984).

In addition to finding the cumulative error we assess the sensitivity of the methods to perturbations of the data. Following Luo et al. (2006), we calculate a running variance of the CPE given by

$$\sigma^2_{\mathsf{RV}} = \frac{1}{n}\sum_{i=1}^{n} CPE_i^2 - \left(\frac{1}{n}\sum_{i=1}^{n} CPE_i\right)^2 \tag{78}$$

and then define a parameter τ ranging over $[0, \sigma_{\mathsf{RV}}]$. Then we perturb our data by forming a new stream $y_i' = y_i + \eta_i$, where each $\eta_i \sim \mathcal{N}(0, \tau^2)$. The η_i's are drawn independently and let us recalculate the cumulative risks as a function of τ for each of the 13 methods. In our graphs below we call these 'sensitivity' curves and denote them by $CPE(\tau)$. In practice, we approximate sensitivity curves on a finite grid in $[0, \sigma_{RV}]$ as shown in Subsec. 7.2.

Our 14 methods are listed in Table 2 together with the abbreviation we use in our graphs to follow and references for where the exact form of the predictor is given in this paper.

Table 2

List of the methods we computed. The only methods that we could run without using a representative subset (indicated by 'rep') were Sht, DPP, Mean, Med, and LSTM. The Shtarkov normal predictor is the same whether 'σ' is known or not and was simply the mean for this case. GPP's and DPP's are the simplest nonparametric Bayes techniques. The hash based procedures are intrinsically one pass. Conformal is Bayes because of the choice of nonconformity. The LSTM's use one layer of 20 levels. Blanks in the right hand column indicate repetitions from the previous line

Method; Class	Abbreviation	Reference
Shtarkov normal; Shtarkov	Sht	(36), $\mu = 0$, $\sigma = 1$
Shtarkov normal; Shtarkov	Sht_rep	
GPP, no random bias; Bayes	GPPnoRB (rep)	(4)
GPP, with random bias; Bayes	GPPRB (rep)	(10), Subsec. 2.1.2
GPP, independent bias; Bayes	GPP_INID (rep)	(18), Subsec. 2.1.3, $\gamma_{n+1} = \bar{\gamma}_n$
Dirichlet; Bayes	DPP	(20)
Dirichlet rep; Bayes	DPP_rep	
Weighted mean; Hash based	Mean	(68)
Weighted mean rep.; Hash based	Mean_rep	
Weighted median; Hash based	Med	(69)
Weighted median rep.; Hash based	Med_rep	
Conformal Bayes; conformal	Conf (rep)	Sec. 5.1, cf. Bersson and Hoff (2024)
LSTM; NN	LSTM	Subsec. 6.3; extension of (76)
LSTM rep; NN	LSTM_rep	

It will be noted that we did not include time series in our computational comparisons. Here's why: we did not want to include frequentist time series because $\mathcal{M}$-open data simply does not have a sampling distribution. Bayesian time series would be philosophically consistent with $\mathcal{M}$-open data invoking the Chen (1985) interpretation. However, the modeling aspect of Bayesian time series should make them perform poorly on $\mathcal{M}$-open data where by definition bias is always a problem. A finding to the contrary would simply mean that the data was probably accurately modeled and hence not $\mathcal{M}$-open, cf. our discussion below of the Accelerometer data. We allowed traditional GPP's and DPP's because they were nonparametric even if they, too, would be subject to bias. On the other hand, we have included LSTM's because they are a currently popular modeling technique used for sequential data. In general, being a model, they perform roughly as one would expect on $\mathcal{M}$-open data, not particularly well.

As can be seen, the methods we compared segregate into two classes. First, some of them are effective for streaming data because the running time per prediction does not increase with stream length. This is the case for Sht, DPP, Med, Mean, and LSTM.

Second, some predictors – the three GPP methods and conformal – have a running time per iteration that increases with stream length. Specifically, for all GPP methods the variance matrix increases in dimension and there is no accepted streaming method to estimate it. Also, for conformal prediction we used the fabContinuousPrediction package in **R** and we did not rewrite the code to make it one pass. Accordingly, we could not realistically use these methods 'as is'. So, we used a 'representative set' in place of the full stream. Our representative set is based on streaming K-means but any streaming clustering procedure could be used. The strategy is to choose a large value of K and use the cluster centers as a our representative set. The representative set has a fixed cardinality, K, but the set itself updates upon receipt of each new y_n. In addition, we used the representative set with Sht, DPP, Mean, and Med, to see if using a representative set made their predictions better or worse.

7.2. Results

To compare empirical performances of the 14 methods, we ran them on six data sets. These data sets have sample sizes that are small even relative to the storage capacity of contemporary laptops. So, our point here is not to argue that one pass methods are essential because of storage constraints or that controlling'running times of sequential predictors is important. We think both of these points are generally accepted. Thus, our point is to compare point predictors from conceptually disjoint predictor classes for extremely complex data.

The sensitivity curves from three data sets are presented here; sensitivity curves from the other three graphs are in Appendix D.2. We chose the three data sets for this section because the median hash based methods give the best performance for the first and the two GPP based methods were best for the second. These two classes of methods seemed to be better than the other three classes of methods; see Subsec. 7.3. The third data set is actually $\mathcal{M}$-complete as we discuss below, being essentially a designed experiment. It shows that methods designed for $\mathcal{M}$-open data can be outperformed by model based methods for complex $\mathcal{M}$-complete data sets.

The first data set, Walmart Sales, see Gibin (2023), has $n = 200000$ and measurements for 6 variables in its columns. It contains data on the unit price of products in Walmart and their quantities sold over 2017-2020. We extracted the first 10000 rows of the data set for our computation where we multiplied

"Quantity" by "UnitPrice" to get the total price and used it. In our graphs we used $\sigma_{RV} = 154$ and a grid with with interval length 19. The sensitivity curves as a function of τ for the 14 methods are plotted in Fig. 7. The values at $\tau = 0$ show the actual CPE for the unperturbed i.e., original, data.

As a generality, we prefer sensitivity curves that are low for τ at or near zero and rise slowly as τ increases. The tradeoff between the value at zero and the rate of increase is hard to assess.

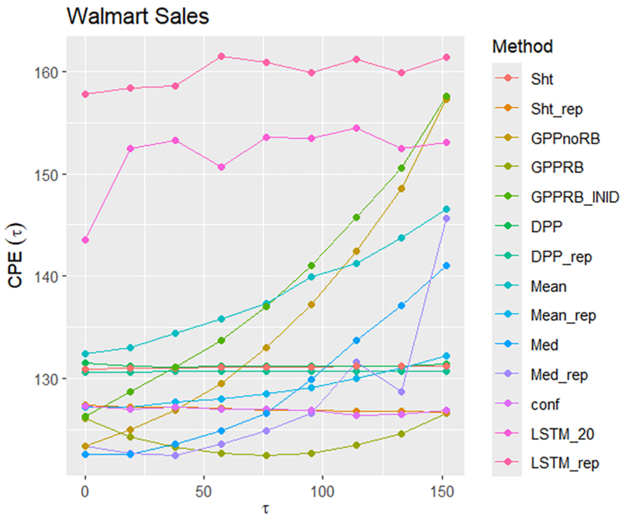

FIG 7. *Sensitivity curves for the* Walmart Sales *data, i.e., cumulative predictive error as a function of the perturbation $CPE(\tau)$. The two median methods perform best and the sensitivity curves for five methods flatline. The other methods give intermediate performance.*

The curves in Fig. 7 show that the medians (representative and one pass) have the lowest error for $\tau = 0$ and are generally rising to the right at a moderate rate compared to the other curves. The mean curves (representative and one-pass) rise as desired but are higher indicating worse CPE. The LSTM curves are arguably the worst performer. The three GPP sensitivity curves either have higher initial error or rise too fast. The curves for five methods – Sht, Sht_rep, Conformal, DPP, and DPP_rep – flatline indicating insensitivity to τ, a highly undesirable property. We interpret this to mean that the predictors represented by these curves are too simple to accommodate the complexity of the data. This has sometimes been called the 'bailout' effect where a predictor (or model selection) procedure defaults to a triviality.

Notably, the GPPRB curve decreases slowly with τ before rising slowly; this indicates that the method performs better when the data have small perturbation than when it has zero perturbation. We regard this as bad but are unclear what it indicates. One conjecture is that there is a variance bias tradeoff dependent on the value of τ.[5] That is, since the data are $\mathcal{M}$-open, there is always bias. So, adding a small perturbation smooths rough data decreasing the error.

However, as τ increases the increased variance dominates any decrease in bias. A curious fact is that to date, we have only seen this with GPPRB here. However, it is a common occurrence in the evaluation of stability in other settings.

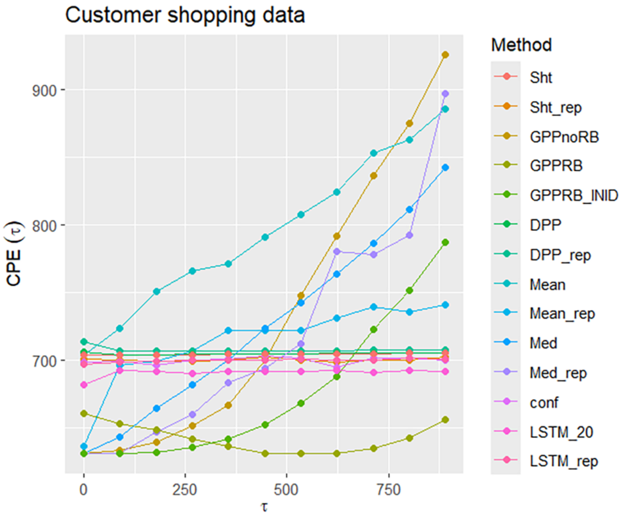

Fig 8. *Sensitivity curves for the* Customer Shopping *data in Istanbul. Two of the GPP methods (GPPnoRB and GPP_INID) performed best along with Med_rep The same five methods as in Fig. 7 flatlined as did LSTM's, essentially. Other methods were intermediate.*

The second data set, Customer Shopping, see Aslan (2023), has customer age, gender, product categories, quantity, price, etc. from 10 different shopping malls in Istanbul between 2021 and 2023. This data set contains 10 columns and like other data sets we have extracted the first 10000 rows from the 99457 rows of the data. For our computational purposes here we have used the column "price". For this data set we found $\sigma_{RV} = 890$ and used a grid with interval length 89 to assess sensitivity.

It is seen from Fig. 8 that GPPRB_INID and GPPnoRB performed best with a close competition from the median_rep. The median one pass method and mean_rep perform a little worse mainly because their sensitivity curves rise quickly as τ increases from zero. Again, the GPPnoRB decreases initially and then increases, both slowly. The mean curve does quite poorly. The same five methods as in Fig. 7 (Sht, Sht_rep, Conformal, DPP, and DPP_rep) flatline as does LSTM. The best performing methods were almost the reverse of Fig. 7, in terms of classes of predictors.

The third data set, Accelerometer, can be found at Sampaio et al. (2019a), which provides a description. The data were presented, analyzed, and assessed in Sampaio et al. (2019b). This is data from a designed experiment that, by construction, is particularly suited to NN methods. For Fig. 9, we used the first 10,000 rows of the data set and used the column "y" for our computing. For

[5] This interpretation is due to Snigdhansu Chatterjee.

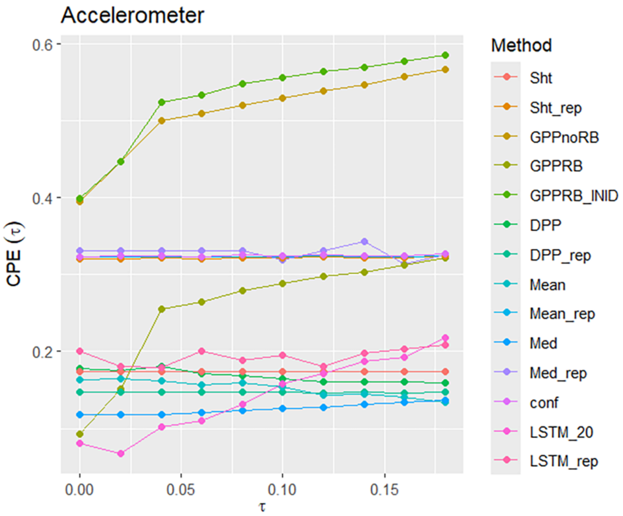

Fig 9. *Sensitivity curves for the* Accelerometer *data, an engineering data set. It is seen that most methods flatline. The LSTM curve has the smallest error at $\tau = 0$ and rises gradually as τ increases. It is noticeably better than the second best curve, GPPRB_INID, that has similar properties. The other curves, starting around $CPE = .4$ are even worse.*

the HBP computations with this data set we used $d_K = 25$, $W_K = 50$, and $K = 200$. To generate the sensitivity curves, we used $\sigma_{RV} = .18$ with τ spaced at interval length of 0.02.

In this case the data are probably $\mathcal{M}$-complete rather than $\mathcal{M}$-open. So, it is not surprising that the LSTM_20 curve (i.e., LSTM's with 20 units) – which is a model – is easily the best performing. For comparison, we redid the computing with 50 units and qualitatively got the same results except that the curves were lower indicating better performance – if only slightly. Interestingly, the second-best curve is GPPINID but is relatively unstable. Two other curves have a very high CPE and the others flatline at one level or another.

Note that a method that is fundamentally model-based is performing well when it is reasonable to assume there is a true model and that predictors that do not incorporate that assumption perform worse. We also see that using a representative set can substantially worsen performance.

7.3. Recommendations

It is difficult to compare the performance of methods in any general sense because it is impossible at this time to produce a systematic comparison of the methods over the class of $\mathcal{M}$-open data sets. Indeed, it is unclear what important and well-defined subclasses of $\mathcal{M}$-open data sets would be. Thus, the observations here are strictly anecdotal.

Consequently, we simply looked at what we thought the top methods were for each data set and scored each method one point if it was a top performer and zero otherwise. This was purely heuristic and subjective. We ruled out methods that flatlined because we regarded them as insensitive to the data. We also ruled out methods that had sensitivity curves that rose too sharply, started at too high a value, or decreased and then increased. We only included the five data sets we argue are $\mathcal{M}$-open; two in the present subsection and three in Appendix D.2. In the 'strictly FWIW' category, we produced Table 3.

TABLE 3

Informal counts of which methods perform best. Only methods with a positive score are shown; all other methods had count zero. Honorable mentions include Mean for the data set Real Estate *but this would not change the informal conclusions*

method	Med	Med Rep	GPP no RB	GPP RB	GPP INID
counts	4	1	2	1	2

It is seen that the only two classes of methods that were ever best in our $\mathcal{M}$-open examples, i.e., excluding LSTM's in Accelerometer, were median hash function based methods and GPP's of one sort or another. This does not mean that all other methods – DPP, conformal, Shtarkov – should be discarded because we only used the simplest conformal and Shtarkov methods and DPP's are arguably the simplest of nonparametric Bayes procedures. Moreover, we see that as a generality, using a representative set did not obviously impair performance of the methods – all GPP methods use one. Although it is unclear what the representative set represents, and other choices of representative sets are possible, it seems using such a set can speed computation with essentially no cost in performance. Recalling that K-means effectively restricts clusters to convex sets this is surprising. On the other hand, we have only computed a few one-dimensional examples so the convexity restriction may be trivial.

8. Discussion

The 14 methods we have presented represent five classes of predictive techniques: Bayes, Shtarkov, hash function based, conformal, and neural network. All of them have been examined in the context of $\mathcal{M}$-open data. Our computations suggest that Bayes methods (GPP's in particular) and methods based on hash functions are likely to perform overall better than the other three classes of methods. This conclusion is reasonable and, with hindsight, perhaps not surprising since they avoid making distributional assumptions on the data.

There are at least two caveats with this summary. First, the concept of $\mathcal{M}$-open is very broad and encompasses data sets with diverse properties. Indeed, we conjecture that as study of this class of data proceeds subsets within the class of $\mathcal{M}$-open data will be defined by their differing properties.

Second, the sensitivity curves of some of the GPP based methods sometimes decreased and then increased indicating that it gave better predictions for small perturbations than for zero perturbations or large perturbations. It remains unclear what this means. If we were modeling, we would conjecture that the model was wrong but that a better model could be found if some aspect like bias could be reduced. That way we might be able to ensure that we would get low error initially and gradually increasing error as the perturbations increased for an improved model. We note that even for $\mathcal{M}$-open data one can use models and even if all of them are wrong some are worse than others.

A clear implication of the material we have presented here is that probability based methods remain fundamental to prediction for $\mathcal{M}$-open data. However, the way probability modeling should be done changes: we do not use probability modeling for the DG, we only use it for the construction of the predictor. This is consistent with the Bayesian view on how to use pre-experimental information.

Probability modeling is used intrinsically in three conceptual classes of techniques (Bayes, hash-based, and Shtarkov) and used indirectly in the other two (LSTM's and CP). However, because we have focused on $\mathcal{M}$-open data sets we cannot use probability modeling in the usual way. We have never posited a distribution for the data; we have only used probability to construct or evaluate the predictors. Summarizing:

1. Bayesian techniques are reinterpreted as in Chen (1985), so we do not have a probability distribution on the data.
2. Hash based methods have a probability only on the choice of hash functions in the predictor; this is like a Bayesian's prior.
3. Shtarkov predictors use probabilistic experts but the DG is not assumed to be probabilistic.
4. Conformal methods are fully empirical and only bring in probability as a way to assess performance. Here, the nonconformity is derived from a posterior density but this is not a distribution on the data.
5. NN's, and LSTM's in particular, are an optimization. They only bring in probability as an error structure for estimation and evaluation.

No survey such as this can be truly comprehensive and, arguably, the biggest conceptual omission is recommender systems. These are not strictly speaking predictors because they give ranks rather than predictions but the highest ranked object could be taken as a prediction. Recommender systems are mainly if not exclusively for discrete data and it is unclear how to adapt them to continuous streams. There is relatively little theoretical development of recommender systems, however, they typically use statistical quantities such as correlations and conditional probabilities (in the form of association rules) without assuming the usual error structure in statistical contexts. The interested reader is referred to the now-classic texts Aggarwal (2016) and Ricci et al. (2010).

Appendix A: Bayes predictors

In this subsection, we give the results needed for the method of moments arguments used in the first two subsections of Sec. 2.

A.1. Method of moments for IID bias in GPP's

Here we give the calculations used in Subsec. 2.1.2. We start with the first moment condition. To show (13), recall from (6) that

$$Y^n | a^n, \sigma^2 \sim \mathcal{N}(a^n, \sigma^2 (I+K)_{n\times n}) \tag{79}$$

and so

$$Y'^{\,n} = (I+K)^{\frac{1}{2}} Y^n \sim \mathcal{N}(a^n, \sigma^2 I_{n\times n}). \tag{80}$$

Also from (7), we have

$$a^n \sim \mathcal{N}(\gamma 1^n, \sigma^2 \delta^2 I_{n\times n}). \tag{81}$$

If we define

$$S_2' \quad = \quad \frac{1}{n-1}\sum_{i=1}^{n}(Y_i' - \bar{Y}')^2,$$

then

$$\begin{aligned} \mathsf{E}(S_2') \quad &= \quad \mathsf{E}\left[\frac{1}{n-1}\sum_{i=1}^{n}(Y_i' - \bar{Y}')^2\right] \\ &= \quad \mathsf{E}\left[\frac{1}{n-1}\left(\sum_{i=1}^{n} Y_i'^2 - n\bar{Y}'^2\right)\right]. \end{aligned} \tag{82}$$

Now, from

$$\mathsf{E}(Y_i') = \mathsf{E}\mathsf{E}(Y_i' | a_i, \sigma^2) = \gamma, \tag{83}$$

we get

$$\begin{aligned} \mathsf{E}(\bar{Y}') = \mathsf{E}\mathsf{E}(\bar{Y}' | a^n, \sigma^2) \quad &= \quad \mathsf{E}\left[\frac{1}{n}\sum_{i=1}^{n}\mathsf{E}(Y_i | a_i, \sigma^2)\right] \\ &= \quad \mathsf{E}\left[\frac{1}{n}\sum_{i=1}^{n} a_i\right] = \gamma. \end{aligned}$$

We also have the identity

$$\mathsf{E}(Y_i'^{\,2}) = \mathsf{Var}(Y_i') + \mathsf{E}^2(Y_i'). \tag{84}$$

Now

$$\begin{aligned} \mathsf{Var}(Y_i') \quad &= \quad \mathsf{E}\mathsf{Var}(Y_i' | a_i, \sigma^2) + \mathsf{Var}\mathsf{E}(Y_i' | a_i, \sigma^2) \\ &= \quad \mathsf{E}(\sigma^2) + \mathsf{Var}(a_i | \sigma^2) = \sigma^2 + \sigma^2\delta^2. \end{aligned} \tag{85}$$

From (83), (84) and, (85), we have

$$\mathsf{E}(Y_i'^2) \quad = \quad \sigma^2 + \sigma^2\delta^2 + \gamma^2. \tag{86}$$

and we recall the identity

$$\mathsf{E}(\bar{Y}^2) \quad = \quad \mathsf{Var}(\bar{Y'}) + \mathsf{E}^2(\bar{Y'}). \tag{87}$$

So, it follows that

$$\begin{aligned}
\mathsf{Var}(\bar{Y'}) &= \mathsf{Var}\mathsf{E}(\bar{Y'}|a^n, \sigma^2) + \mathsf{E}\mathsf{Var}(\bar{Y'}|a^n, \sigma^2) \\
&= \mathsf{Var}\left(\frac{1}{n}\sum_{i=1}^{n} a_i|a^n, \sigma^2\right) + \mathsf{E}\mathsf{Var}\left(\frac{1}{n}\sum_{i=1}^{n} Y_i'|a^n, \sigma^2\right) \\
&= \frac{1}{n^2}\sum_{i=1}^{n} \mathsf{Var}(a_i) + \mathsf{E}\left(\frac{1}{n^2}\sum_{i=1}^{n} \mathsf{Var}(Y_i')|a^n, \sigma^2\right) \\
&= \frac{\sigma^2\delta^2}{n} + \mathsf{E}\left(\frac{1}{n^2} n\sigma^2\right) \\
&= \frac{\sigma^2\delta^2}{n} + \frac{\sigma^2}{n}.
\end{aligned} \tag{88}$$

Hence from (84), (87) and (88), we have

$$\mathsf{E}(\bar{Y'}^2) \quad = \quad \frac{\sigma^2\delta^2}{n} + \frac{\sigma^2}{n} + \gamma^2. \tag{89}$$

Then from (82), (86), and (89) we have

$$\begin{aligned}
\mathsf{E}(S_2') &= \frac{1}{n-1}\left[\sum_{i=1}^{n}(\sigma^2 + \sigma^2\delta^2 + \gamma^2) - n\left(\frac{\sigma^2\delta^2}{n} + \frac{\sigma^2}{n} + \gamma^2\right)\right] \\
&= \sigma^2(1+\delta^2).
\end{aligned}$$

Hence,

$$\mathsf{E}\left(\frac{S_2'}{1+\delta^2}\right) \quad = \quad \sigma^2. \tag{90}$$

So, setting $\hat{\sigma}^2 = \frac{S_2'}{1+\delta^2}$ we have unbiasedness.

For a second moment condition to use in the method of moments argument in Subsec. 2.1.2, we find $Var(\hat{\sigma}^2)$. We know that if $X \sim \mathcal{N}(\mu, \sigma^2)$, then

$$\mathsf{E}(X^4) = \mu^4 + 6\mu^2\sigma^2 + 3\sigma^4 \tag{91}$$

Using this with Y', we start with

$$\mathsf{E}(\bar{Y'}^4) \quad = \quad \mathsf{E}\mathsf{E}(\bar{Y'}^4|a^n, \sigma^2).$$

From (80), we can say that

$$\bar{Y'}|a^n, \sigma^2 \sim \mathcal{N}(\bar{a^n}, \frac{\sigma^2}{n})$$

$$\mathsf{E}(\bar{Y'}^4|a^n, \sigma^2) \quad = \quad \bar{a}^4 + 6\bar{a}^2\frac{\sigma^2}{n} + 3\frac{\sigma^4}{n^2} \tag{92}$$

From (81), we have,

$$\bar{a} \sim \mathcal{N}(\gamma, \frac{\sigma^2\delta^2}{n}).$$

We have

$$\begin{aligned} \mathsf{E}(\bar{a}^2) &= \mathsf{Var}(\bar{a}) + \mathsf{E}^2(\bar{a}) \\ &= \frac{\sigma^2\delta^2}{n} + \gamma^2. \end{aligned}$$

Also from (91), we can say that,

$$\mathsf{E}(\bar{a}^4) \quad = \quad \gamma^4 + 6\frac{\gamma^2\sigma^2\delta^2}{n} + \frac{3}{n^2}\sigma^4\delta^4. \tag{93}$$

Hence from (92) and (93), we get

$$\begin{aligned} \mathsf{E}(\bar{Y'}^4) &= \mathsf{E}(\bar{Y'}^4|a^n, \sigma^2) \\ &= \mathsf{E}(\bar{a}^4) + 6\frac{\sigma^2}{n}\mathsf{E}(\bar{a}^2) + \frac{3}{n^2}\sigma^4. \\ &= \gamma^4 + \frac{6}{n}\gamma^2\sigma^2\delta^2 + \frac{3}{n^2}\sigma^4\delta^4 + \frac{6}{n^2}\sigma^4\delta^2 + \frac{6}{n}\sigma^2\gamma^2 + \frac{3}{n^2}\sigma^4. \end{aligned} \tag{94}$$

Now,

$$\begin{aligned} \mathsf{E}(S_2'^2) &= \mathsf{E}\left[\left\{\frac{1}{n-1}\sum_{i=1}^{n}(Y_i' - \bar{Y'})^2\right\}^2\right] \\ &= \frac{1}{(n-1)^2}\Bigg[\sum_{i=1}^{n}\mathsf{E}(Y_i'^2)\sum_{j=1}^{n}\mathsf{E}(Y_j'^2) - n\mathsf{E}(\bar{Y'}^2)\sum_{i=1}^{n}\mathsf{E}(Y_i'^2) \\ &\quad -n\mathsf{E}(\bar{Y'}^2)\sum_{j=1}^{n}\mathsf{E}(Y_j'^2) + n^2\mathsf{E}(\bar{Y'}^4)\Bigg] \end{aligned} \tag{95}$$

Then from (86), (89), and (94), we have

$$\begin{aligned} \mathsf{E}(S_2'^2) &= \frac{1}{(n-1)^2}\Bigg[n^2(\sigma^2 + \sigma^2\delta^2 + \gamma^2)^2 - 2n^2(\sigma^2 + \sigma^2\delta^2 + \gamma^2) \\ &\quad \left\{\gamma^2 + \frac{1}{n}(\sigma^2 + \sigma^2\delta^2)\right\} + n^2\Bigg(\gamma^4 + \frac{6}{n}\gamma^2\sigma^2\delta^2 + \frac{3}{n^2}\sigma^4\delta^4 + \frac{6}{n^2}\sigma^4\delta^2 \end{aligned}$$

$$
\begin{aligned}
& +\frac{6}{n}\sigma^2\gamma^2+\frac{3}{n^2}\sigma^4\Big)\Big]\\
= \;& \frac{1+\delta^2}{(n-1)^2}[\sigma^4(n^2-2n+3)(1+\delta^2)+4n\sigma^2\gamma^2]. \qquad (96)
\end{aligned}
$$

Hence,

$$
\mathsf{Var}\left(\frac{S_2'}{1+\delta^2}\right) \;=\; \frac{1}{(1+\delta^2)^2}[\mathsf{E}(S_2'^{\,2})-\mathsf{E}^2(S_2')].
$$

From (82) and (96), we have

$$
\begin{aligned}
Var\left(\frac{S_2'}{1+\delta^2}\right)=Var(\hat{\sigma}^2)&=\frac{1}{(n-1)^2}[\sigma^4(n^2-2n+3)+\frac{4n}{1+\delta^2}\sigma^2\gamma^2]-\sigma^4\\
&=\frac{2\sigma^2}{(n-1)^2}\left[\sigma^2+\frac{2n\gamma^2}{1+\delta^2}\right]. \qquad (97)
\end{aligned}
$$

A.2. Proof of Theorem 2.2

This proof has been adapted from Chanda et al. (2026). The terms that do not contain γ in the proof have not been changed.

We start by noting that the joint prior distribution is given by

$$
\begin{aligned}
w(a^n,\sigma^2) &= \mathcal{N}(\gamma 1^n,\sigma^2\delta^2 I_{n\times n})\;\mathcal{IG}(\alpha,\beta)\\
&= \frac{e^{-\frac{1}{2\sigma^2}(a^n-\gamma^n)'(\delta^2 I_{n\times n})^{-1}(a^n-\gamma^n)}}{(2\pi)^{\frac{n}{2}}(\sigma^2\delta^2)^{\frac{n}{2}}}\frac{\beta^\alpha}{\Gamma(\alpha)}\left(\frac{1}{\sigma^2}\right)^{\alpha+1}e^{-\frac{\beta}{\sigma^2}}.
\end{aligned}
$$

Set,

$$
\begin{aligned}
\mu^{\gamma n} &= [(I+K)^{-1}_{n\times n}+(\delta^2 I_{n\times n})^{-1}]^{-1}[(I+K)^{-1}_{n\times n}y^n+(\delta^2 I_{n\times n})^{-1}\gamma^n]\\
&= V_{n\times n}[(I+K)^{-1}_{n\times n}y^n+\gamma(\delta^2 I_{n\times n})^{-1}1^n]\\
\beta_n^{*\gamma} &= \beta+\frac{1}{2}y'^{\,n}(I+K)^{-1}_{n\times n}y^n+\frac{1}{2}\gamma^{n'}[\delta^2 I_{n\times n}]^{-1}\gamma^n-\frac{1}{2}\mu^{\gamma' n}V^{-1}_{n\times n}\mu^{\gamma n}\\
&= \beta+\frac{1}{2}y'^{\,n}[(I+K)^{-1}_{n\times n}-(I+K)^{-1}_{n\times n}V_{n\times n}(I+K)^{-1}_{n\times n}]y^n\\
&\quad -\frac{1}{\delta^2}y'^{\,n}(I+K)^{-1}_{n\times n}V_{n\times n}\gamma^n+\frac{n}{2}\frac{\gamma^{n'}\gamma^n}{\delta^2}-\frac{1}{2}\frac{1}{\delta^4}\gamma'^{\,n}V_{n\times n}\gamma^n.
\end{aligned}
$$

We have, $w(a^n,\sigma^2|y^n)=\mathcal{L}_1(y^n|a,\sigma^2)\times w(a^n,\sigma^2)$ Then,

$$
\begin{aligned}
w(a^n,\sigma^2|y^n) &= \frac{\beta^\alpha}{(2\pi)^n|I+K|^{\frac{1}{2}}(\delta^2)^{\frac{n}{2}}\Gamma(\alpha)}\left(\frac{1}{\sigma^2}\right)^{\alpha^*+1}\\
&\quad \times e^{-\frac{1}{\sigma^2}[\frac{1}{2}(a^n-\mu^{\gamma n})'V^{-1}_{n\times n}(a^n-\mu^{\gamma n})]}e^{-\frac{1}{\sigma^2}\beta^{*\gamma}}
\end{aligned}
$$

$$
m(y^n) \;=\; \frac{|V_{n\times n}|^{\frac{1}{2}}}{(2\pi\delta^2)^{\frac{n}{2}}|I+K|^{\frac{1}{2}}}\frac{\beta^\alpha}{\Gamma(\alpha)}\frac{\Gamma(\alpha^*-\frac{n}{2})}{\beta_n^{*\gamma^{\alpha^*-\frac{n}{2}}}}.
$$

Then,

$$\frac{m(y^{n+1})}{m(y^n)} \quad = \quad c\frac{(\beta^{*\gamma}_{n+1})^{-(\alpha+\frac{n+1}{2})}}{(\beta^{*\gamma}_{n})^{-(\alpha+\frac{n}{2})}} \tag{98}$$

where, c is defined in Chanda et al. (2026). Generalizing (98) for $(n+1)$ observations, we get,

$$\begin{aligned}\beta^{\gamma *}_{n+1} \quad = \quad & \beta + \frac{1}{2}y'^{\,n+1}[(I+K)^{-1}_{n+1\times n+1} \\ & -(I+K)^{-1}_{n+1\times n+1}V_{n+1\times n+1}(I+K)^{-1}_{n+1\times n+1}]y^{n+1} \\ & -\frac{1}{\delta^2}y'^{\,n+1}(I+K)^{-1}_{n+1\times n+1}V_{n+1\times n+1}\gamma^{n+1} + \frac{n+1}{2}\frac{\gamma'^{\,n+1}\gamma^{n+1}}{\delta^2} \\ & -\frac{1}{2}\frac{1}{\delta^4}\gamma'^{\,n+1}V_{n+1\times n+1}\gamma^{n+1}.\end{aligned} \tag{99}$$

Redefine

$$\begin{aligned}\Gamma_2^{\gamma,n+1} \quad &= \quad \frac{1}{\delta^2}(I+K)^{-1}_{n+1\times n+1}V_{n+1\times n+1}\gamma^{n+1} \\ \Delta^{\gamma} \quad &= \quad \frac{n+1}{2}\frac{\gamma^2}{\delta^2} - \frac{1}{2}\frac{\gamma^2}{\delta^4}1'^{\,n+1}V_{n+1\times n+1}1^n\end{aligned}$$

$$\beta^{\gamma *}_{n+1} \quad = \quad \beta + \frac{1}{2}y'^{\,n+1}\Gamma_{1,n+1\times n+1}y^{n+1} - y'^{\,n+1}\Gamma_2^{\gamma,n+1} + \Delta^{\gamma}. \tag{100}$$

, where $\Gamma_{1,n+1\times n+1}$ has been defined in the Festschrift paper.

Now, we partition y^{n+1}, $\Gamma_{1,n+1\times n+1}$, and Γ_2^{n+1}. Write

$$\begin{aligned}y'^{\,n+1}\Gamma_{1,n+1\times n+1}y^{n+1} \quad &= \quad \begin{pmatrix} y^n & y_{n+1}\end{pmatrix}\left[\begin{array}{c|c} \Gamma_{1,n\times n} & g_1^{*n} \\ \hline g_1'^{\,*n} & g_1^* \end{array}\right]\begin{pmatrix} y^n \\ y_{n+1}\end{pmatrix} \\ &= \quad y'^{\,n}\Gamma_{1,n\times n}y^n + 2y'^{\,n}g_1^{n*}y_{n+1} + y_{n+1}^2 g_1^*\end{aligned} \tag{101}$$

and

$$y'^{\,n+1}\Gamma_2^{n+1,\gamma} \quad = \quad \begin{pmatrix} y^n & y_{n+1}\end{pmatrix}\begin{pmatrix}\Gamma_2^{n,\gamma} \\ g_2^*\end{pmatrix} = y'^{\,n}\Gamma_2^{n,\gamma} + y_{n+1}g_2^*. \tag{102}$$

Using (101) and (102) in (100), we get

$$\begin{aligned}\beta^{*,\gamma}_{n+1} \quad &= \quad \beta + \frac{1}{2}y'^{\,n}\Gamma_{1,n\times n}y^n - y'^{\,n}\Gamma_2^{n,\gamma} + \Delta^{\gamma} + \frac{1}{2}g_1^*y_{n+1}^2 - y_{n+1}(g_2^* - y'^{\,n}g_1^{n*}) \\ &= \quad \beta + \frac{g_1^*}{2}(y_{n+1} - A_1^*)^2 + A_2^*.\end{aligned} \tag{103}$$

where,

$$A_1^* \quad = \quad \frac{g_2^* - y'^{\,n}g_1^{*n}}{g_1^*}$$

$$A_2^* \quad = \quad \frac{1}{2} y'^n \Gamma_{1,n\times n} y^n - y'^n \Gamma_2^{n,\gamma} + \Delta^\gamma - \frac{1}{2g_1^*}(g_2^* - y'^n g_1^{*n})^2. \tag{104}$$

Now, since $m(y^n)$ is the marginal density of y^n and, $m(y^{n+1})$ is the marginal density of y^{n+1},

$$\int_{\mathbb{R}} \frac{m(y^{n+1})}{m(y^n)} \mathrm{dy_{n+1}} = 1.$$

From (98) we have that

$$\int_{\mathbb{R}} c \times \frac{{\beta_{n+1}^{*,\gamma}}^{-\left(\alpha+\frac{n+1}{2}\right)}}{\beta_n^{*,\gamma-\left(\alpha+\frac{n}{2}\right)}} \mathrm{dy_{n+1}} \quad = \quad 1. \tag{105}$$

So solving for c gives

$$c \quad = \quad \frac{\beta_n^{*,\gamma-\left(\alpha+\frac{n}{2}\right)}}{\int_{\mathbb{R}} {\beta_{n+1}^{*,\gamma}}^{-\left(\alpha+\frac{n+1}{2}\right)} \mathrm{dy_{n+1}}}.$$

Using (105) in (98), we have

$$\frac{m(y^{n+1})}{m(y^n)} \quad = \quad \frac{{\beta_{n+1}^{*,\gamma}}^{-\left(\alpha+\frac{n+1}{2}\right)}}{\int_{\mathbb{R}} {\beta_{n+1}^{*,\gamma}}^{-\left(\alpha+\frac{n+1}{2}\right)} \mathrm{dy_{n+1}}}. \tag{106}$$

Rename, $\beta^{**,\gamma} = \beta + A_2^*$.

$${\beta_{n+1}^{*,\gamma}}^{-(\alpha+\frac{n+1}{2})} \quad = \quad \beta^{**,\gamma-\left(\alpha+\frac{n}{2}\right)} \beta^{**,\gamma-\frac{1}{2}} \left[1 + \frac{g_1^*}{2\beta^{**,\gamma}}(y_{n+1} - A_1^*)^2\right]^{-\left(\alpha+\frac{n+1}{2}\right)} \tag{107}$$

By definition, the t-density is given by

$$St_v(\tau, \Sigma)(g) \quad = \quad \frac{\Gamma(\frac{v+d}{2})}{\Gamma(\frac{v}{2})\pi^{\frac{d}{2}}|v\Sigma|^{\frac{1}{2}}} \left(1 + \frac{(g-\tau)'\Sigma^{-1}(g-\tau)}{v}\right)^{-\frac{v+d}{2}}. \tag{108}$$

So if we let

$$v = 2\alpha, d = 1, \Sigma = \frac{\beta^{**\gamma}}{\frac{2\alpha+n}{2}} \frac{1}{g_1^*}, g = y_{n+1}, \tau = A_1^*. \tag{109}$$

and use (109) in (108), we get

$$\begin{aligned} St_{2\alpha+n}\left(A_1^*, \frac{\beta^{**,\gamma}}{\frac{2\alpha+n}{2}}\right)(y_{n+1}) \quad &= \quad \frac{\Gamma(\frac{2\alpha+n+1}{2})}{\Gamma(\frac{2\alpha+n}{2})} {g_1^*}^{\frac{1}{2}} \frac{1}{(2\pi)^{\frac{1}{2}}} \beta^{**,\gamma-\frac{1}{2}} \\ &\quad \times \left[1 + \frac{g_1^*}{2\beta^{**,\gamma}}(y_{n+1} - A_1^*)^2\right]^{-\frac{2\alpha+n+1}{2}}. \end{aligned}$$

Hence,

$$\beta^{**,\gamma-\frac{1}{2}}\left[1+\frac{g_1^*}{2\beta^{**,\gamma}}(y_{n+1}-A_1^*)^2\right]^{-\left(\alpha+\frac{n+1}{2}\right)}$$
$$=\frac{\Gamma(\frac{2\alpha+n+1}{2})}{\Gamma(\frac{2\alpha+n}{2})}\frac{(2\pi)^{\frac{1}{2}}}{g_1^{*\,\frac{1}{2}}}\times St_{2\alpha+n}\left(A_1,\frac{\beta^{**}}{\frac{2\alpha+n}{2}}\right)(y_{n+1}). \tag{110}$$

Using (110) in (107), and (107) in (106), we have

$$\frac{m(y^{n+1})}{m(y^n)} = \frac{\beta^{**,\gamma-(\alpha+\frac{n}{2})}}{\int_{\mathbb{R}}\beta_{n+1}^{*,\gamma}{}^{-(\alpha+\frac{n+1}{2})}\mathrm{dy_{n+1}}}\frac{\Gamma(\frac{2\alpha+n}{2})}{\Gamma(\frac{2\alpha+n+1}{2})}\frac{(2\pi)^{\frac{1}{2}}}{(g_1^*)^{\frac{1}{2}}}$$
$$\times St_{2\alpha+n}\left(A_1,\frac{\beta^{**,\gamma}}{\frac{2\alpha+n}{2}}\right)(y_{n+1}). \tag{111}$$

Since $\frac{m(y^{n+1})}{m(y^n)}=m(y_{n+1}|y^n)$ is a density, $\int_{\mathbb{R}}\frac{m(y^{n+1})}{m(y^n)}\mathrm{dy_{n+1}}=1$. Integrating the right hand side of (111) w.r.t y_{n+1} gives that

$$\frac{\beta^{**,\gamma-(\alpha+\frac{n}{2})}}{\int_{\mathbb{R}}\beta_{n+1}^{*,\gamma}{}^{-(\alpha+\frac{n+1}{2})}\mathrm{dy_{n+1}}}\frac{\Gamma(\frac{2\alpha+n}{2})}{\Gamma(\frac{2\alpha+n+1}{2})}\frac{(2\pi)^{\frac{1}{2}}}{(\gamma_1)^{\frac{1}{2}}}\int_{\mathbb{R}}St_{2\alpha+n}\left(A_1,\frac{\beta^{**,\gamma}}{\frac{2\alpha+n}{2}}\right)(y_{n+1})\mathrm{dy_{n+1}} \tag{112}$$

equals 1, since y_{n+1} is only in the argument of the t distribution. The integral of the t distribution being one means (112) gives

$$\frac{\beta^{**,\gamma-(\alpha+\frac{n}{2})}}{\int_{\mathbb{R}}\beta_{n+1}^{*,\gamma}{}^{-(\alpha+\frac{n+1}{2})}\mathrm{dy_{n+1}}}\frac{\Gamma(\frac{2\alpha+n}{2})}{\Gamma(\frac{2\alpha+n+1}{2})}\frac{(2\pi)^{\frac{1}{2}}}{(g_1^*)^{\frac{1}{2}}}=1. \tag{113}$$

Finally using (113) in (111), we get

$$m(y_{n+1}|y^n) = St_{2\alpha+n}\left(A_1,\frac{\beta^{**,\gamma}}{\frac{2\alpha+n}{2}}\right)(y_{n+1}). \quad \square$$

A.3. Method of moments for non-identical biases in GPP's

From (17), we have,

$$a^n\sim\mathcal{N}(\gamma^n,\sigma^2\delta^2 I_{n\times n}).$$

Hence,

$$\bar{a}\sim\mathcal{N}(\bar{\gamma}=\frac{1}{n}\sum_{i=1}^{n}\gamma_i,\frac{\sigma^2\delta^2}{n}).$$

From (80),

$$E(Y_i')=\gamma_i. \tag{114}$$

and

$$E(\bar{Y'}) = \bar{\gamma}. \tag{115}$$

Hence from (114) and (85), we have

$$E({Y_i'}^2) = \sigma^2\delta^2 + \sigma^2 + \gamma_i^2. \tag{116}$$

From (115) and (88), we have

$$E(\bar{Y'}^2) = \frac{\sigma^2}{n}(1+\delta^2) + \bar{\gamma}^2. \tag{117}$$

From (82), (116), and (117), we have

$$\begin{aligned} E(S_2') &= \frac{1}{n-1}\left[\sum_{i=1}^{n}(\sigma^2\delta^2 + \sigma^2 + \gamma_i^2) - n\left\{\frac{\sigma^2}{n}(1+\delta^2) + \bar{\gamma}^2\right\}\right] \\ &= \sigma^2(1+\delta^2) + \frac{1}{n-1}\sum_{i=1}^{n}(\gamma_i - \bar{\gamma})^2. \end{aligned} \tag{118}$$

Define $S_2^\gamma = \frac{1}{n-1}\sum_{i=1}^{n}(\gamma_i - \bar{\gamma})^2$. Then, (118) becomes

$$\mathsf{E}(S_2') = \sigma^2(1+\delta^2) + S_2^\gamma. \tag{119}$$

Then

$$\mathsf{E}\left(\frac{S_2' - S_2^\gamma}{1+\delta^2}\right) = \sigma^2.$$

Define

$$\hat{\sigma}^2 = \frac{S_2' - S_2^\gamma}{1+\delta^2}. \tag{120}$$

$$\mathsf{Var}\left(\frac{S_2' - S_2^\gamma}{1+\delta^2}\right) = \frac{\mathsf{Var}(S_2')}{(1+\delta^2)^2}. \tag{121}$$

We can show that

$$\mathsf{E}(\bar{Y'}^4) = \bar{\gamma}^4 + \frac{6}{n}\bar{\gamma}^2\sigma^2\delta^2 + \frac{3}{n^2}\sigma^4\delta^4 + \frac{3}{n^2}\sigma^4 + \frac{6}{n^2}\sigma^4\delta^2 + \frac{6}{n}\sigma^2\bar{\gamma}^2. \tag{122}$$

From (95), (116), (89), and (122), we have

$$\begin{aligned} \mathsf{E}({S_2'}^2) = &\frac{1}{(n-1)^2}\left[\sum_{i=1}^{n}(\sigma^2\delta^2 + \sigma^2 + \gamma_i^2)\sum_{j=1}^{n}(\sigma^2\delta^2 + \sigma^2 + \gamma_j^2)\right. \\ &-n\left\{\frac{\sigma^2}{n}(1+\delta^2) + \bar{\gamma}^2\right\}\sum_{i=1}^{n}(\sigma^2\delta^2 + \sigma^2 + \gamma_i^2) \end{aligned}$$

$$
\begin{aligned}
&-n\left\{\frac{\sigma^2}{n}(1+\delta^2)+\bar{\gamma}^2\right\}\sum_{j=1}^{n}(\sigma^2\delta^2+\sigma^2+\gamma_j^2)\\
&+n^2\left(\bar{\gamma}^4+\frac{6}{n}\bar{\gamma}^4+\frac{3}{n^2}\sigma^4\delta^4+\frac{3}{n^2}\sigma^4+\frac{6}{n^2}\sigma^4\delta^2+\frac{6}{n}\sigma^2\bar{\gamma}^2\right)\bigg]\\
= &\frac{1}{(n-1)^2}[\sigma^4(n^2-2n+3)(1+\delta^2)^2+2\sigma^2(n-1)(1+\delta^2)\sum_{i=1}^{n}\gamma_i^2\\
&-2n(n-3)\sigma^2\bar{\gamma}^2(1+\delta^2)+\big(\sum_{i=1}^{n}\gamma_i^2-n\bar{\gamma}^2\big)^2]\\
= &\frac{1}{(n-1)^2}[\sigma^4(n^2-2n+3)(1+\delta^2)^2+2\sigma^2(n-1)(1+\delta^2)\sum_{i=1}^{n}\gamma_i^2\\
&-2n(n-3)\sigma^2\bar{\gamma}^2(1+\delta^2)]+S_2^{\gamma 2}. \qquad (123)
\end{aligned}
$$

From (118) and (123) (121) becomes

$$
\begin{aligned}
\mathsf{Var}\left(\frac{S_2'}{1+\delta^2}\right) &= \frac{1}{(1+\delta^2)^2}\left[\frac{1}{(n-1)^2}\{\sigma^4(n^2-2n+3)(1+\delta^2)^2\right.\\
&\quad +2\sigma^2(n-1)(1+\delta^2)\sum_{i=1}^{n}\gamma_i^2-2n(n-3)\sigma^2\bar{\gamma}^2(1+\delta^2)\}\\
&\quad \left.+(n-1)^2S_2^{\gamma 2}-\{\sigma^2(1+\delta^2)+S_2^{\gamma}\}^2\right]\\
&= \sigma^2\left[\frac{2\sigma^2}{(n-1)^2}-\frac{2}{1+\delta^2}S_2^{\gamma}\right.\\
&\quad \left.+\frac{2}{n-1}\frac{1}{1+\delta^2}\left(\sum_{i=1}^{n}\gamma_i^2-\frac{n(n-3)}{n-1}\bar{\gamma}^2\right)\right]\\
&= \frac{2\sigma^2}{(n-1)^2}\left[\sigma^2+\frac{2n\bar{\gamma}^2}{1+\delta^2}\right]. \qquad (124)
\end{aligned}
$$

Since (15) and (16) continue to hold in the INID case, we can use (120) and (124) to get $\hat{\alpha}$ and $\hat{\beta}$.

Appendix B: Proofs from Sec. 3.1

Proof of Theorem 3.2:

First we observe that the MLE, resp. MPLE, exists and is unique under the first two conditions. Hence it is enough to show that the denominators $D_{n,\mathsf{F}}$ and $D_{n,\mathsf{B}}$ are finite and bounded away from zero for fixed n.

We begin with $D_{n,\mathsf{F}}$. First, we see $\int_{y^n} p(y^n|\hat{\theta})dy^n < \infty$: Under conditions 1, 3, and 4 there is an $R > 0$ so that $p(y|\theta) < R$ and, $|y| < R$. Now,

$$\begin{aligned}\int_{y^n} p(y^n|\hat{\theta})dy^n &= \int p(y^n|\hat{\theta})\chi_{|y_i|<R}dy^n \\ &\leq \int R\chi_{|y_i|<R}dy^n \leq R(2R)^n < \infty.\end{aligned}$$

Next, we show $\int p(y^n|\hat{\theta})\mathrm{d}y^n > 0$. Regard $\hat{\theta}$ as a function $\hat{\theta} : \mathcal{Y}^n \to \Theta$ and denote the interior of the image of $\hat{\theta}$ by $Im^0(\hat{\theta})$. For any $\theta_0 \in Im^0(\hat{\theta})$ there must be a y_0^n for which $\hat{\theta}(y_0^n) = \theta_0$. For this y_0^n, $\exists \tau > 0$ so that $p(y_0^n|\theta_0) > \tau$: by contradiction, if no such τ existed then $p(y_0^n|\hat{\theta}) = 0$ then since $\hat{\theta}$ is a maximum we have that $p(y_0^n|\theta) = 0$ for all θ. Thus, $\hat{\theta}$ is not unique.

Since $p(\cdot|\cdot)$ is continuous in y^n and $\hat{\theta}$, we have that for this $\tau > 0$, there is an $r > 0$ so that for y^n and θ satisfying $|y^n - y_0^n| < r$ and $|\hat{\theta} - \theta_0| < r$ we have $p(y^n|\theta) > \frac{\tau}{2} > 0$. But now, by continuity of $\hat{\theta}$, there exists $r' \leq r$ so that $|y^n - y_0^n| < r'$ implies that $|\hat{\theta} - \theta_0| < r$. Writing $B(r') = \{|y^n - y_0| < r'\}$, we have

$$\begin{aligned}\int_{y^n} p(y^n|\hat{\theta})dy^n &\geq \int_{B(r')} p(y^n|\hat{\theta})dy^n \\ &\geq \frac{\tau}{2}\int_{B(r')} dy^n \\ &= \frac{\tau}{2} \times \text{Volume of a ball in } \mathbb{R}^n \text{of radius } r' \\ &= \frac{\tau}{2}\frac{\pi^{\frac{n}{2}}}{\Gamma(\frac{n}{2}+1)}(r')^n > 0.\end{aligned}$$

Hence, $q_{\mathsf{opt,F}}$ is well-defined.

The proof for $q_{\mathsf{opt,B}}$ is similar. There is an R so that $p(y|\theta)$, $w(\hat{\theta})$, and $|y^n|$ are bounded by R. So, we have that

$$\begin{aligned}\int w(\hat{\theta})p(y^n|\hat{\theta})\mathrm{d}y^n &= \int_{\{|y^n|<R\}} w(\hat{\theta})p(y^n|\hat{\theta})\mathrm{d}y^n \\ &\leq R^2\int_{\{|y^n|<R\}} \mathrm{d}y^n \leq R^2(2R)^n < \infty.\end{aligned}$$

To show $\int_{y^n} w(\tilde{\theta})p(y^n|\tilde{\theta})dy^n > 0$, choose $\theta_0 \in Im^0(\tilde{\theta})$. As before, there is a y_0^n so that $\tilde{\theta}(y_0^n) = \theta_0$ and a $\tau > 0$ so that $w(\theta_0)p(y_0^n|\theta_0) > \tau > 0$. (By way of contradiction, if no such τ exists, $w(\theta_0)p(y_0^n|\theta) = 0$ for all θ so the maximum is not unique.) In fact, we can find $\tau > 0$ so that $w(\theta_0) > \tau$ and $p(y_0^n|\theta_0) > \tau$.

Since $p(\cdot|\cdot)$ is continuous in its arguments, say y^n and θ, there exists $r > 0$ so that $|y^n - y_0^n| < r$ and $|\hat{\theta} - \theta_0| < r$ implies $p(y^n|\tilde{\theta}) > \tau/2 > 0$. Since the prior density is continuous we can also assume that r and τ can be chosen small enough that $w(\tilde{\theta}(y_0^n)) > \tau/2$. But now, by continuity of $\tilde{\theta}$, there exists $r' \leq r$, such that $|y^n - y_0^n| < r' \implies |\tilde{\theta}(y^n) - \theta_0| < r$. So,

$$\begin{aligned}
\int_{y^n} w(\tilde{\theta}) p(y^n|\tilde{\theta}) dy^n &\geq \int_{B(r')} w(\tilde{\theta}) p(y^n|\tilde{\theta}) dy^n \\
&\geq \frac{\tau}{2} \int_{B(r')} w(\tilde{\theta}) dy^n \\
&\geq \frac{\tau^2}{4} \int_{B(r')} \mathrm{d}y^n \\
&= \frac{\tau^2}{4} \text{ Volume of a ball in } R^n \text{ of radius } r' \\
&= \frac{\tau^2}{4} \frac{\pi^{\frac{n}{2}}}{\Gamma(\frac{n}{2}+1)} (r')^n > 0.
\end{aligned}$$

Hence, $q_{\mathsf{opt,B}}$ is well-defined.

Appendix C: Calculus proofs for Shtarkov predictors

This section is to record proofs that are important for the sake of completeness.

C.1. Normal cases

For the mean and variance unknown, we verify that $\hat{y}_{n+1}$ maximizes (32). So, we check if the second derivative of (32) w.r.t. y_{n+1} is negative at $\hat{y}_{n+1}$. Note that

$$\frac{d^2}{dy_{n+1}^2} \ln p(y^{n+1}|\hat{\mu}_{n+1}, \hat{\sigma^2}_{n+1}) = \tag{125}$$

$$-\frac{\{\sum_{i=1}^n y_i^2 + y_{n+1}^2 - \frac{(n\bar{y}_n + y_{n+1})^2}{n+1}\} - (y_{n+1} - \bar{y}_n)\left\{2y_{n+1} - \frac{2(n\bar{y}_n + y_{n+1})}{n+1}\right\}}{[\sum_{i=1}^n y_i^2 + y_{n+1}^2 - \frac{1}{n+1}(n\bar{y}_n + y_{n+1})^2]^2}.$$

The denominator of (125) is strictly positive. So, we will just focus on the numerator. Let us rename it Num. So, the numerator is

$$\begin{aligned}
Num &= -\Bigg[\sum_{i=1}^n y_i^2 + y_{n+1}^2 - \frac{(n\bar{y}_n + y_{n+1})^2}{n+1}\} \\
&\quad -(y_{n+1} - \bar{y}_n)\left\{2y_{n+1} - \frac{2(n\bar{y}_n + y_{n+1})}{n+1}\right\}\Bigg].
\end{aligned} \tag{126}$$

Substituting y_{n+1} with $\hat{y}_{n+1} = \bar{y}_n$ in (126), we have,

$$\begin{aligned}
Num &= -\left[\sum_{i=1}^n y_i^2 + \bar{y}_n^2 - \frac{(n\bar{y}_n + \bar{y}_n)^2}{n+1}\right] \\
&= -\sum_{i=1}^n (y_i - \bar{y}_n)^2 < 0.
\end{aligned} \tag{127}$$

Hence, (125) < 0. So, $\hat{y}_{n+1}$ maximizes (32).

Next we turn to the normal Bayes Shtarkov case for the normal with both μ and σ unknown. Using the identity

$$\begin{aligned} & \sum_{i=1}^{n}(y_i-\mu)^2+\left(\frac{\mu-\mu_0}{\sigma_0}\right)^2 \\ = & \left(n+\frac{1}{\sigma_0^2}\right)\left[\mu-\frac{n\bar{y}+\frac{\mu_0}{\sigma_0^2}}{n+\frac{1}{\sigma_0^2}}\right]^2-\frac{(n\bar{y}+\frac{\mu_0}{\sigma_0^2})^2}{n+\frac{1}{\sigma_0^2}}+\left(\sum_{i=1}^{n}y_i^2+\frac{\mu_0^2}{\sigma_0^2}\right). \end{aligned}$$

in (35) gives

$$\begin{aligned} & p(y^n|\mu,\sigma^2)\times p(\mu|\mu_0,\sigma_0^2)\times p(\sigma^2|\alpha,\beta) \\ \propto & \left(\frac{1}{\sigma^2}\right)^{\frac{n}{2}+\alpha+1}\left(\frac{1}{\sigma^2}\right)^{\frac{1}{2}}e^{-\frac{\beta}{\sigma^2}} \\ & \times e^{-\frac{1}{2\sigma^2}\left[\left(n+\frac{1}{\sigma_0^2}\right)\left\{\mu-\frac{n\bar{y}+\frac{\mu_0}{\sigma_0^2}}{n+\frac{1}{\sigma_0^2}}\right\}^2-\frac{(n\bar{y}+\frac{\mu_0}{\sigma_0^2})^2}{n+\frac{1}{\sigma_0^2}}+\left(\sum_{i=1}^{n}y_i^2+\frac{\mu_0^2}{\sigma_0^2}\right)\right]} \\ = & \left(\frac{1}{\sigma^2}\right)^{\frac{n}{2}+\alpha+1}\left(\frac{1}{\sigma^2}\right)^{\frac{1}{2}}e^{-\frac{1}{\sigma^2}\left[\beta+\frac{1}{2}\left(\sum_{i=1}^{n}y_i^2+\frac{\mu_0^2}{\sigma_0^2}\right)-\frac{1}{2}\frac{(n\bar{y}+\frac{\mu_0}{\sigma_0^2})^2}{n+\frac{1}{\sigma_0^2}}\right]} \\ & \times e^{-\frac{1}{2\sigma^2}\frac{n\sigma_0^2+1}{\sigma_0^2}\left(\mu-\frac{n\bar{y}+\frac{\mu_0}{\sigma_0^2}}{n+\frac{1}{\sigma_0^2}}\right)^2}. \end{aligned} \tag{128}$$

Hence,

$$\mu|\sigma^2,\sigma_0^2,\mu_0 \sim \mathcal{N}\left(\frac{n\bar{y}+\frac{\mu_0}{\sigma_0^2}}{n+\frac{1}{\sigma_0^2}},\frac{\sigma^2\sigma_0^2}{n\sigma_0^2+1}\right) \tag{129}$$

$$\sigma^2|\alpha,\beta \sim \mathcal{IG}\left(\alpha+\frac{n}{2},\beta+\frac{1}{2}\left\{\sum_{i=1}^{n}y_i^2+\frac{\mu_0^2}{\sigma_0^2}-\frac{(n\bar{y}+\frac{\mu_0}{\sigma_0^2})^2}{n+\frac{1}{\sigma_0^2}}\right\}\right). \tag{130}$$

Thus, we have

$$\hat{\mu}_{\mathsf{MPLE}} = \frac{n\bar{y}+\frac{\mu_0}{\sigma_0^2}}{n+\frac{1}{\sigma_0^2}} \tag{131}$$

$$\hat{\sigma^2}_{\mathsf{MPLE}} = \frac{\beta+\frac{1}{2}\left\{\sum_{i=1}^{n}y_i^2+\frac{\mu_0^2}{\sigma_0^2}-\frac{(n\bar{y}+\frac{\mu_0}{\sigma_0^2})^2}{n+\frac{1}{\sigma_0^2}}\right\}}{\alpha+\frac{n}{2}+1}. \tag{132}$$

Replacing μ by $\hat{\mu}_{MPLE}$ and σ^2 by $\hat{\sigma}^2_{MPLE}$ for $n+1$ copies of y in (128) and using $\bar{y}_{n+1}=(n\bar{y}+y_{n+1})/(n+1)$ gives

$$\begin{aligned} & p(y^{n+1}|\hat{\mu}_{n+1,\mathsf{MPLE}},\hat{\sigma^2}_{n+1,\mathsf{MPLE}})\times p(\hat{\mu}_{n+1,\mathsf{MPLE}}|\mu_0,\sigma_0^2) \\ & \times p(\hat{\sigma^2}_{n+1,\mathsf{MPLE}}|\alpha,\beta) \end{aligned}$$

$$
\begin{aligned}
\propto \quad & \left[\frac{\alpha+\frac{n+1}{2}+1}{\beta+\frac{1}{2}\left\{\sum_{i=1}^{n} y_i^2+y_{n+1}^2+\frac{\mu_0^2}{\sigma_0^2}-\frac{\left\{(n+1)\frac{1}{n+1}(n\bar{y}_n+y_{n+1})+\frac{\mu_0}{\sigma_0^2}\right\}^2}{n+1+\frac{1}{\sigma_0^2}}\right\}}\right]^{\alpha+\frac{n+2}{2}+1} \\
& \times e^{-\frac{(\alpha+\frac{n+1}{2}+1)\left[\beta+\frac{1}{2}\left\{\sum_{i=1}^{n} y_i^2+y_{n+1}^2+\frac{\mu_0^2}{\sigma_0^2}-\frac{\left\{(n+1)\frac{1}{n+1}(n\bar{y}_n+y_{n+1})+\frac{\mu_0}{\sigma_0^2}\right\}^2}{n+1+\frac{1}{\sigma_0^2}}\right\}\right]}{\beta+\frac{1}{2}\left\{\sum_{i=1}^{n} y_i^2+y_{n+1}^2+\frac{\mu_0^2}{\sigma_0^2}-\frac{\left\{(n+1)\frac{1}{n+1}(n\bar{y}_n+y_{n+1})+\frac{\mu_0}{\sigma_0^2}\right\}^2}{n+1+\frac{1}{\sigma_0^2}}\right\}}} \\
& \times e^{-\frac{\frac{(n+1)\bar{y}_{n+1}+\frac{\mu_0}{\sigma_0^2}}{n+1+\frac{1}{\sigma_0^2}}-\frac{(n+1)\bar{y}_{n+1}+\frac{\mu_0}{\sigma_0^2}}{n+1+\frac{1}{\sigma_0^2}}}{2\hat{\sigma}^2_{n+1,MPLE}}}. \\
= \quad & \left[\frac{\alpha+\frac{n+1}{2}+1}{\beta+\frac{1}{2}\left\{\sum_{i=1}^{n} y_i^2+y_{n+1}^2+\frac{\mu_0^2}{\sigma_0^2}-\frac{\left\{(n\bar{y}_n+y_{n+1})+\frac{\mu_0}{\sigma_0^2}\right\}^2}{n+1+\frac{1}{\sigma_0^2}}\right\}}\right]^{\alpha+\frac{n+2}{2}+1} \\
& \times e^{-(\alpha+\frac{n+1}{2}+1)}. \qquad (133)
\end{aligned}
$$

Taking logarithms on both sides of (133), we get

$$
\begin{aligned}
& \ln[p(y^{n+1}|\hat{\mu}_{n+1,\mathsf{MPLE}}, \hat{\sigma^2}_{n+1,\mathsf{MPLE}}) \times p(\hat{\mu}_{n+1,\mathsf{MPLE}}|\mu_0,\sigma_0^2) \\
& \times p(\hat{\sigma^2}_{n+1,\mathsf{MPLE}}|\alpha,\beta)] \\
& \propto \left(\alpha+\frac{n+2}{2}+1\right)\ln\left(\alpha+\frac{n+1}{2}+1\right)-\left(\alpha+\frac{n+2}{2}+1\right) \\
& \times \ln\left[\beta+\frac{1}{2}\left\{\sum_{i=1}^{n} y_i^2+y_{n+1}^2+\frac{\mu_0^2}{\sigma_0^2}-\frac{\left\{(n\bar{y}_n+y_{n+1})+\frac{\mu_0}{\sigma_0^2}\right\}^2}{n+1+\frac{1}{\sigma_0^2}}\right\}\right] \\
& -\left(\alpha+\frac{n+1}{2}+1\right). \qquad (134)
\end{aligned}
$$

Differentiating both sides of (134) w.r.t y_{n+1} and equating to 0 gives

$$
\begin{aligned}
\frac{-\left(\alpha+\frac{n+2}{2}+1\right)\times\frac{1}{2}\left[2y_{n+1}-\frac{2\left(n\bar{y}_n+y_{n+1}+\frac{\mu_0}{\sigma_0^2}\right)}{n+1+\frac{1}{\sigma_0^2}}\right]}{\beta+\frac{1}{2}\left\{\sum_{i=1}^{n} y_i^2+y_{n+1}^2+\frac{\mu_0^2}{\sigma_0^2}-\frac{\left\{(n\bar{y}_n+y_{n+1})+\frac{\mu_0}{\sigma_0^2}\right\}^2}{n+1+\frac{1}{\sigma_0^2}}\right\}} &= 0 \\
\implies y_{n+1}-\frac{2\left(n\bar{y}_n+y_{n+1}+\frac{\mu_0}{\sigma_0^2}\right)}{n+1+\frac{1}{\sigma_0^2}} &= 0 \\
\implies \hat{y}_{n+1} &= \frac{n\bar{y}_n+\frac{\mu_0}{\sigma_0^2}}{n+\frac{1}{\sigma_0^2}}
\end{aligned}
$$

as the point predictor as claimed.

To verify it is a maximum, we differentiate

$$\begin{aligned}&\frac{d}{dy_{n+1}}\ln[p(y^{n+1}|\hat{\mu}_{n+1,\mathsf{MPLE}},\hat{\sigma^2}_{n+1,\mathsf{MPLE}})\times p(\hat{\mu}_{n+1,\mathsf{MPLE}}|\mu_0,\sigma_0^2)\\&\quad\times p(\hat{\sigma^2}_{n+1,\mathsf{MPLE}}|\alpha,\beta)]\end{aligned}$$

w.r.t. y_{n+1} to get

$$\begin{aligned}&\frac{d^2}{dy_{n+1}^2}\ln[p(y^{n+1}|\hat{\mu}_{n+1,\mathsf{MPLE}},\hat{\sigma^2}_{n+1,\mathsf{MPLE}})\times p(\hat{\mu}_{n+1,\mathsf{MPLE}}|\mu_0,\sigma_0^2)\\&\quad\times p(\hat{\sigma^2}_{n+1,\mathsf{MPLE}}|\alpha,\beta)]\\ = \;& \frac{-(\alpha+\frac{n}{2}+2)}{\left[\beta+\frac{1}{2}\left\{\sum_{i=1}^n y_i^2+y_{n+1}^2+\frac{\mu_0}{\sigma_0^2}-\frac{(n\bar{y}_n+y_{n+1}+\frac{\mu_0}{\sigma_0^2})^2}{n+1+\frac{1}{\sigma_0^2}}\right\}\right]^2}\left[\left(1-\frac{1}{n+1+\frac{1}{\sigma_0^2}}\right)\right.\\&\times\left\{\beta+\frac{1}{2}\left(\sum_{i=1}^n y_i^2+y_{n+1}^2+\frac{\mu_0}{\sigma_0^2}-\frac{(n\bar{y}_n+y_{n+1}+\frac{\mu_0}{\sigma_0^2})^2}{n+1+\frac{1}{\sigma_0^2}}\right)\right\}\\&\left.-\left(y_{n+1}-\frac{n\bar{y}_n+y_{n+1}+\frac{\mu_0}{\sigma_0^2}}{n+1+\frac{1}{\sigma_0^2}}\right)\left\{\frac{1}{2}\left(2y_{n+1}-2\frac{(n\bar{y}_n+y_{n+1}+\frac{\mu_0}{\sigma_0^2})}{n+1+\frac{1}{\sigma_0^2}}\right)\right\}\right].\end{aligned}$$

Renaming, we set

$$\begin{aligned}A &= \frac{-(\alpha+\frac{n}{2}+2)}{\left[\beta+\frac{1}{2}\left\{\sum_{i=1}^n y_i^2+y_{n+1}^2+\frac{\mu_0}{\sigma_0^2}-\frac{(n\bar{y}_n+y_{n+1}+\frac{\mu_0}{\sigma_0^2})^2}{n+1+\frac{1}{\sigma_0^2}}\right\}\right]^2}\\ B &= \left(1-\frac{1}{n+1+\frac{1}{\sigma_0^2}}\right)\left[\beta+\frac{1}{2}\left\{\sum_{i=1}^n y_i^2+y_{n+1}^2+\frac{\mu_0}{\sigma_0^2}-\frac{(n\bar{y}_n+y_{n+1}+\frac{\mu_0}{\sigma_0^2})^2}{n+1+\frac{1}{\sigma_0^2}}\right\}\right]\\ C &= \left(y_{n+1}-\frac{n\bar{y}_n+y_{n+1}+\frac{\mu_0}{\sigma_0^2}}{n+1+\frac{1}{\sigma_0^2}}\right)\left[\frac{1}{2}\left\{2y_{n+1}-2\frac{(n\bar{y}_n+y_{n+1}+\frac{\mu_0}{\sigma_0^2})}{n+1+\frac{1}{\sigma_0^2}}\right\}\right].\end{aligned}$$

Now,

$$\begin{aligned}&\frac{d^2}{dy_{n+1}^2}\ln[p(y^{n+1}|\hat{\mu}_{n+1,MPLE},\hat{\sigma^2}_{n+1,MPLE})\times p(\hat{\mu}_{n+1,MPLE}|\mu_0,\sigma_0^2)\\&\quad\times p(\hat{\sigma^2}_{n+1,MPLE}|\alpha,\beta)]\\ =\;& \text{Term A(Term B - Term C)}. \qquad (135)\end{aligned}$$

If we put $\hat{y}_{n+1}$ in each of Term A, Term B, and Term C we get the following expressions. First, the easiest one is

$$\text{Term C} = \left(\hat{y}_{n+1}-\frac{n\bar{y}_n+y_{n+1}+\frac{\mu_0}{\sigma_0^2}}{n+1+\frac{1}{\sigma_0^2}}\right)\left[\frac{1}{2}\left\{2\hat{y}_{n+1}-2\frac{(n\bar{y}_n+\hat{y}_{n+1}+\frac{\mu_0}{\sigma_0^2})}{n+1+\frac{1}{\sigma_0^2}}\right\}\right]$$

$$
\begin{aligned}
&= \left(\hat{y}_{n+1} - \frac{n\bar{y}_n + \hat{y}_{n+1} + \frac{\mu_0}{\sigma_0^2}}{n+1+\frac{1}{\sigma_0^2}}\right)^2 \\
&= \frac{n\hat{y}_{n+1} + \hat{y}_{n+1} + \frac{y_{n+1}}{\sigma_0^2} - n\bar{y}_n - \hat{y}_{n+1} - \frac{\mu_0}{\sigma_0^2}}{n+1+\frac{1}{\sigma_0^2}} \\
&= \frac{\hat{y}_{n+1}(n+\frac{1}{\sigma_0^2}) - (n\bar{y}_n + \frac{\mu_0}{\sigma_0^2})}{n+1+\frac{1}{\sigma_0^2}} \\
&= \frac{\frac{n\bar{y}_n+\frac{\mu_0}{\sigma_0^2}}{n+\frac{1}{\sigma_0^2}}(n+\frac{1}{\sigma_0^2}) - (n\bar{y}_n + \frac{\mu_0}{\sigma_0^2})}{n+1+\frac{1}{\sigma_0^2}} = 0. \qquad (136)
\end{aligned}
$$

Hence, (135) becomes

$$
\begin{aligned}
&\frac{d^2}{dy_{n+1}^2}\ln[p(y^{n+1}|\hat{\mu}_{n+1,MPLE},\hat{\sigma^2}_{n+1,MPLE}) \times p(\hat{\mu}_{n+1,MPLE}|\mu_0,\sigma_0^2) \\
&\quad \times p(\hat{\sigma^2}_{n+1,MPLE}|\alpha,\beta)] \\
&= \text{Term A Term B.}
\end{aligned}
$$

More explicitly, the product of A and B is

$$
\begin{aligned}
&\frac{-(\alpha+\frac{n}{2}+2)}{\left[\beta+\frac{1}{2}\left\{\sum_{i=1}^n y_i^2 + y_{n+1}^2 + \frac{\mu_0}{\sigma_0^2} - \frac{(n\bar{y}_n+y_{n+1}+\frac{\mu_0}{\sigma_0^2})^2}{n+1+\frac{1}{\sigma_0^2}}\right\}\right]^2}\frac{n+\frac{1}{\sigma_0^2}}{n+1+\sigma_0^2} \\
&\times\left[\beta+\frac{1}{2}\left\{\sum_{i=1}^n y_i^2 + y_{n+1}^2 + \frac{\mu_0}{\sigma_0^2} - \frac{(n\bar{y}_n+y_{n+1}+\frac{\mu_0}{\sigma_0^2})^2}{n+1+\frac{1}{\sigma_0^2}}\right\}\right] \\
= \; &\frac{-(\alpha+\frac{n}{2}+2)\frac{n+\frac{1}{\sigma_0^2}}{n+1+\sigma_0^2}}{\beta+\frac{1}{2}\left\{\sum_{i=1}^n y_i^2 + y_{n+1}^2 + \frac{\mu_0}{\sigma_0^2} - \frac{(n\bar{y}_n+y_{n+1}+\frac{\mu_0}{\sigma_0^2})^2}{n+1+\frac{1}{\sigma_0^2}}\right\}}.
\end{aligned}
$$

So, at $\hat{y}_{n+1}$,

$$
\begin{aligned}
&\frac{d^2}{dy_{n+1}^2}\ln[p(y^{n+1}|\hat{\mu}_{n+1,\mathsf{MPLE}},\hat{\sigma^2}_{n+1,\mathsf{MPLE}}) \times p(\hat{\mu}_{n+1,\mathsf{MPLE}}|\mu_0,\sigma_0^2) \\
&\quad \times p(\hat{\sigma^2}_{n+1,\mathsf{MPLE}}|\alpha,\beta)] \\
= \; &\frac{-(\alpha+\frac{n}{2}+2)\frac{n+\frac{1}{\sigma_0^2}}{n+1+\sigma_0^2}}{\beta+\frac{1}{2}\left\{\sum_{i=1}^n y_i^2 + \hat{y}_{n+1}^2 + \frac{\mu_0}{\sigma_0^2} - \frac{(n\bar{y}_n+\hat{y}_{n+1}+\frac{\mu_0}{\sigma_0^2})^2}{n+1+\frac{1}{\sigma_0^2}}\right\}} \\
= \; &\frac{-(\alpha+\frac{n}{2}+2)\frac{n+\frac{1}{\sigma_0^2}}{n+1+\sigma_0^2}}{\beta+\frac{1}{2}\left\{\sum_{i=1}^n y_i^2 + (\frac{n\bar{y}_n+\frac{\mu_0}{\sigma_0^2}}{n+\frac{1}{\sigma_0^2}})^2 + \frac{\mu_0}{\sigma_0^2} - \frac{(n\bar{y}_n+\frac{n\bar{y}_n+\frac{\mu_0}{\sigma_0^2}}{n+\frac{1}{\sigma_0^2}}+\frac{\mu_0}{\sigma_0^2})^2}{n+1+\frac{1}{\sigma_0^2}}\right\}}
\end{aligned}
$$

$$= \quad \frac{-(\alpha + \frac{n}{2} + 2)\frac{n+\frac{1}{\sigma_0^2}}{n+1+\sigma_0^2}}{\beta + \frac{1}{2}\left\{\sum_{i=1}^n y_i^2 - \frac{(n\bar{y}_n + \frac{\mu_0}{\sigma_0^2})^2}{n+\frac{1}{\sigma_0^2}} + \frac{\mu_0}{\sigma_0^2}\right\}}. \tag{137}$$

By definition, if y_{n+1} is a maximum, it will maximize the joint likelihood function

$$p(y^{n+1}|\hat{\mu}_{n+1,MPLE}, \hat{\sigma}^2{}_{n+1,MPLE}) \times p(\hat{\mu}_{n+1,MPLE}|\mu_0, \sigma_0^2) \times p(\hat{\sigma}^2{}_{n+1,MPLE}|\alpha, \beta).$$

For simplicity, we only look at the case $\mu_0 = 0$ and $\sigma_0 = 1$. Then, (137) becomes

$$\begin{aligned} &\frac{d^2}{dy_{n+1}^2} \ln[p(y^{n+1}|\hat{\mu}_{n+1,\mathsf{MPLE}}, \hat{\sigma}^2{}_{n+1,\mathsf{MPLE}}) \times p(\hat{\mu}_{n+1,\mathsf{MPLE}}) \\ &\quad \times p(\hat{\sigma}^2{}_{n+1,\mathsf{MPLE}}|\alpha, \beta)] \\ = \quad & -\left(\frac{n+1}{n+2}\right) \frac{\alpha + \frac{n}{2} + 2}{\beta + \frac{1}{2}(\sum_{i=1}^n y_i^2 - \frac{n^2}{n+1}\bar{y}_n^2)}. \end{aligned} \tag{138}$$

Since

$$\begin{aligned} \sum_{i=1}^n y_i^2 - \frac{n^2}{n+1}\bar{y}_n^2 \quad &= \quad \sum_{i=1}^n y_i^2 - n\bar{y}_n^2 + n\bar{y}_n^2 - \frac{n^2}{n+1}\bar{y}_n^2 \\ &= \quad (n-1)\mathsf{Var}(Y) + \frac{n}{n+1}\bar{y}_n^2 > 0, \end{aligned}$$

we see that (138) < 0. Other values of μ_0 and σ_0 are similar.

C.2. Binomial cases

For the beta-binomial case, the derivation for (52) is the following. Multiply (50) and (51) to get

$$\begin{aligned} p(y^n|\theta) \times w(\theta|\alpha, \beta) \quad &= \quad \left\{\prod_{i=1}^n \binom{N}{y_i}\right\} \theta^{\sum_{i=1}^n y_i}(1-\theta)^{Nn - \sum_{i=1}^n y_i} \\ &\quad \times \frac{1}{\mathcal{B}(\alpha, \beta)}\theta^{\alpha-1}(1-\theta)^{\beta-1} && (139) \\ &\propto \quad \theta^{\alpha + \sum_{i=1}^n y_i - 1}(1-\theta)^{\beta + Nn - \sum_{i=1}^n y_i - 1} \\ &= \quad \theta^{\alpha + n\bar{y}_n - 1}(1-\theta)^{\beta + Nn - n\bar{y}_n - 1}. && (140) \end{aligned}$$

Hence, $\theta|y^n, \alpha, \beta \sim \mathcal{B}(\alpha + n\bar{y}_n, \beta + Nn - n\bar{y}_n)$ and

$$\hat{\theta}_{\mathsf{MPLE}} = \frac{\alpha + n\bar{y}_n - 1}{\alpha + \beta + Nn - 2}. \tag{141}$$

Substituting (141) into (139), we have

$$p(y^n|\hat{\theta}_{\mathsf{MPLE}})w(\hat{\theta}_{\mathsf{MPLE}}, \alpha, \beta)$$

$$= \frac{1}{\mathcal{B}(\alpha,\beta)}\left\{\prod_{i=1}^{n}\binom{N}{y_i}\right\}\left[\frac{\alpha+n\bar{y}_n-1}{\alpha+\beta+Nn-2}\right]^{\alpha+n\bar{y}_n-1} \times\left[\frac{\beta+Nn-n\bar{y}_n-1}{\alpha+\beta+Nn-2}\right]^{\beta+Nn-n\bar{y}_n-1}. \quad (142)$$

So, for $y_1, y_2, \cdots, y_n, y_{n+1}$, we can rewrite (142) as:

$$\begin{aligned} & p(y^{n+1}|\hat{\theta}_{\mathsf{MPLE}})w(\hat{\theta}_{\mathsf{MPLE}},\alpha,\beta) \\ & \propto \binom{N}{y_{n+1}}(\alpha+n\bar{y}_n+y_{n+1}-1)^{\alpha+n\bar{y}_n+y_{n+1}-1} \\ & \times(\beta+N(n+1)-n\bar{y}_n-y_{n+1}-1)^{\beta+N(n+1)-n\bar{y}_n-y_{n+1}-1}. \end{aligned} \quad (143)$$

Taking logarithms on both sides of (143) we get (52).

C.3. Gamma cases

Taking the logarithm on both sides of (55), we get

$$\ln p_\alpha(y^{n+1}|\hat{\theta}_{\mathsf{MLE}}) \quad \propto \quad (\alpha-1)\ln y_{n+1}-(n+1)\alpha\ln\left(\sum_{i=1}^{n}y_i+y_{n+1}\right). \quad (144)$$

Differentiating (144), w.r.t. y_{n+1} and setting it to 0 gives

$$\hat{y}_{n+1} \quad = \quad \frac{n(\alpha-1)\bar{y}_n}{n\alpha+1}. \quad (145)$$

To be sure this $\hat{y}_{n+1}$ is a maximum we look at three ranges of α. If $\alpha = 1$, we return to the exponential case and can see the result directly from the maximized likelihood. For $\alpha > 1$, we verify the second derivative of (144) at $\hat{y}_{n+1}$ is negative. We have

$$\frac{d^2}{dy_{n+1}^2}\ln p_\alpha(y^{n+1}|\hat{\theta}_{\mathsf{MLE}}) = \frac{y_{n+1}^2(n+1)\alpha+(1-\alpha)(n\bar{y}_n+y_{n+1})^2}{(n\bar{y}_n+y_{n+1})^2 y_{n+1}^2}. \quad (146)$$

The denominator in (146) is strictly positive. So, it is enough to look only at the numerator. Rename the numerator

$$Q(y_{n+1}) = Q_{\bar{y}_n}(y_{n+1}) = y_{n+1}^2(1+n\alpha)+(1-\alpha)(2y_{n+1}n\bar{y}_n+n^2\bar{y}_n^2). \quad (147)$$

Now, replacing y_{n+1} with $\hat{y}_{n+1}$, we get,

$$\begin{aligned} Q(\hat{y}_{n+1}) &= \frac{n^2(\alpha-1)^2\bar{y}_n^2}{(n\alpha+1)^2}(1+n\alpha)+(1-\alpha)\frac{2n(\alpha-1)\bar{y}_n}{n\alpha+1}n\bar{y}_n+(1-\alpha)n^2\bar{y}_n^2 \\ &= -\frac{n^2\bar{y}_n^2}{n\alpha+1}(\alpha-1)\alpha(n+1). \end{aligned} \quad (148)$$

For $\alpha > 1$, (148) is negative. Hence, (55) is maximised at $y_{n+1} = \hat{y}_{n+1}$.

For $\alpha \in (0, 1)$, we can see directly that (55) is maximized at $\hat{y}_{n+1} = 0$. Simply rewrite (55) as

$$p(y^{n+1}|\hat{\theta}_{MLE}, \alpha) \quad \propto \quad \left(\frac{1}{y_{n+1}}\right)^{1-\alpha} \left(\frac{1}{n\bar{y}_n + y_{n+1}}\right)^{(n+1)\alpha}. \tag{149}$$

Now, both terms are maximised at $y_{n+1} = 0$.

For the Bayesian Gamma Shtarkov family of experts we want to show (60). Start by substituting (59) in (58) to get

$$\begin{aligned} & p(y^n|\hat{\theta}_{\mathsf{MPLE}}, \alpha) \times w(\hat{\theta}_{\mathsf{MPLE}}|\alpha_0, \beta_0) \\ & = \frac{\beta_0^{\alpha_0}}{\Gamma(\alpha_0)} \left(\prod_{i=1}^n y_i^{\alpha-1}\right) \frac{1}{\Gamma(\alpha)^n} \left(\frac{n\alpha + \alpha_0 - 1}{n\bar{y}_n + \beta_0}\right)^{n\alpha+\alpha_0-1} \\ & \times e^{-\frac{n\alpha+\alpha_0-1}{n\bar{y}_n+\beta_0}(n\bar{y}_n+\beta_0)} \\ \propto \quad & \left(\prod_{i=1}^n y_i^{\alpha-1}\right) \left(\frac{1}{n\bar{y}_n + \beta_0}\right)^{n\alpha+\alpha_0-1}. \end{aligned} \tag{150}$$

For $n+1$ samples $y_1, y_2, \cdots, y_n, y_{n+1}$ (150) can be rewritten as

$$\begin{aligned} & p(y^{n+1}|\hat{\theta}_{\mathsf{MPLE}}, \alpha) \times w(\hat{\theta}_{\mathsf{MPLE}}|\alpha_0, \beta_0) \\ \propto \quad & \left(\prod_{i=1}^n y_i^{\alpha-1}\right) y_{n+1}^{\alpha-1} \left(\frac{1}{\beta_0 + (n+1)\bar{y}_{n+1}}\right)^{(n+1)\alpha+\alpha_0-1} \\ \propto \quad & y_{n+1}^{\alpha-1} \left(\frac{1}{\beta_0 + n\bar{y}_n + y_{n+1}}\right)^{(n+1)\alpha+\alpha_0-1}. \end{aligned} \tag{151}$$

Taking logarithm on both sides of (151), we get

$$\begin{aligned} & \ln[p(y^{n+1}|\hat{\theta}_{\mathsf{MPLE}}, \alpha) w(\hat{\theta}_{\mathsf{MPLE}}|\alpha_0, \beta_0)] \\ = \quad & (\alpha - 1) \ln y_{n+1} - \{(n+1)\alpha + \alpha_0 - 1\} \\ & \times \ln\left(\beta_0 + n\bar{y}_n + y_{n+1}\right) \end{aligned} \tag{152}$$

Differentiating both sides of (152) w.r.t. y_{n+1}, we have

$$\begin{aligned} & \frac{d}{dy_{n+1}} [\ln p(y^{n+1}|\hat{\theta}_{\mathsf{MPLE}}, \alpha) \times w(\hat{\theta}_{\mathsf{MPLE}}|\alpha_0, \beta_0)] \\ = \quad & \frac{\alpha - 1}{y_{n+1}} - \frac{(n+1)\alpha + \alpha_0 - 1}{\beta_0 + n\bar{y}_n + y_{n+1}}. \end{aligned} \tag{153}$$

Equating (153) to 0 and solving for y_{n+1} gives us (60).

To see that (60) is a maximum, start with $\alpha < 1$. Then,

$$p(y^{n+1}|\hat{\theta}_{\mathsf{MPLE}}, \alpha) \times w(\hat{\theta}_{\mathsf{MPLE}}|\alpha_0, \beta_0)$$

$$\begin{aligned}
&\propto \quad y_{n+1}^{\alpha-1}\left(\frac{1}{\beta_0 + n\bar{y}_n + y_{n+1}}\right)^{(n+1)\alpha+\alpha_0-1}. \\
&= \quad \left(\frac{1}{y_{n+1}}\right)^{1-\alpha}\left(\frac{1}{\beta_0 + n\bar{y}_n + y_{n+1}}\right)^{(n+1)\alpha+\alpha_0-1}. \qquad (154)
\end{aligned}$$

We can see directly that (154) is maximized at $\hat{y}_{n+1} = 0$.

For $\alpha = 1$,

$$\begin{aligned}
&p(y^{n+1}|\hat{\theta}_{\mathsf{MPLE}}, \alpha = 1) \times w(\hat{\theta}_{\mathsf{MPLE}}|\alpha_0, \beta_0) \\
\propto \quad &\left(\frac{1}{y_{n+1}}\right)^{1-1}\left(\frac{1}{\beta_0 + n\bar{y}_n + y_{n+1}}\right)^{(n+1)+\alpha_0-1}. \qquad (155)
\end{aligned}$$

which is also maximum at $\hat{y}_{n+1} = 0$.

For $\alpha > 1$, differentiate (153) w.r.t. y_{n+1} to find

$$\begin{aligned}
&\frac{d^2}{d^2 y_{n+1}} \ln[p(y^{n+1}|\hat{\theta}_{\mathsf{MPLE}}, \alpha) w(\hat{\theta}_{\mathsf{MPLE}}|\alpha_0, \beta_0)] \\
= \quad &\frac{-(\alpha-1)(\beta_0 + n\bar{y}_n + y_{n+1})^2 + [(n+1)\alpha + \alpha_0 - 1]y_{n+1}^2}{y_{n+1}^2(\beta_0 + n\bar{y}_n + y_{n+1})^2} \\
= \quad &\frac{Numerator}{Denominator}. \qquad (156)
\end{aligned}$$

The *Denominator* is positive and the *Numerator* at $\hat{y}_{n+1} = \frac{(\alpha-1)(\beta_0+n\bar{y}_n)}{n\alpha+\alpha_0}$ is

$$\begin{aligned}
&-(\alpha-1)\left[\beta_0 + n\bar{y}_n + \frac{(\alpha-1)(\beta_0 + n\bar{y}_n)}{n\alpha + \alpha_0}\right]^2 \\
&+[(n+1)\alpha + \alpha_0 - 1]\frac{(\alpha-1)^2}{(n\alpha+\alpha_0)^2}(\beta_0 + n\bar{y}_n)^2 \\
= \quad &-(\alpha-1)(n\alpha+\alpha_0)(n\alpha+\alpha_0+\alpha-1)\frac{(\beta_0 + n\bar{y}_n)^2}{(n\alpha+\alpha_0)^2} < 0. \qquad (157)
\end{aligned}$$

Appendix D: Further computational details

D.1. Details of implementation

In this subsection we give the details of implementation of the 13 methods grouped into their four classes as indicated in Table 2. Again, the reader can skip to the results in Subsec. 7.2 if desired. For all of our computed results here, we used a burn-in of 10% of the sample size for each data set. We used the burn-in set to get our first predictor. When we used a representative subset from streaming K-means, we set $K = 200$ and simply took the first 200 distinct points as the initial representative set.

D.1.1. Shtarkov

For the Shtarkov method, we chose our 'experts' to be normal distributions. In all cases where the mean was unknown the Shtarkov predictor was either the mean itself or an affine function of the mean defined by the hyperparameters in the prior. In these cases, the predictor defaulted to the mean for large n This was simple to implement in one-pass or to modify by using a representative set.

D.1.2. Bayes

We used five Bayesian methods. Even though the conformal method was based on the posterior as a nonconformity we did not regard this as a Bayes method because it did not use the posterior directly to get a prediction. Moreover, there are other technique that use a density for a nonconformity that we did not use simply because code was not available. This was a judgment call and readers may prefer to regard this conformal predictor as Bayesian.

We grouped the five Bayesian methods into three that were based on GPP's and two that were based on DPP's.

1. **GPP methods**
   We used three GPP based predictors. For the first, the GPP with no random bias, the predictor is given in equation (4) where the terms K_{11} and K_{12} need to be specified. For the second, the predictor in GPPRB (IID Additive case) is specified in Theorem 2.1 where the form of the predictor has a function A_1 in it. This function depends on other functions g_1, g_2, g_1^n which can be explicitly written as the functions of the hyperparameters γ, δ, the variance matrix $K_{n+1\times n+1}$ and the data. for the third, the predictor in GPPRB (INID Additive bias case) is given by A_1^* where the details about the predictor can be found in Theorem 2.2. For this method the variance matrix $K_{n\times n}$ must be chosen.
   In all cases, we chose the variance matrices to correspond to an $AR(1)$ correlation matrix which has the form

$$K_{n\times n} = \begin{pmatrix} 1 & \rho & \cdots & \rho^{n-1} \\ \rho & 1 & \cdots & \rho^{n-2} \\ \vdots & \vdots & \ddots & \vdots \\ \rho^{n-1} & \rho^{n-2} & \cdots & 1 \end{pmatrix}$$

   and we set $\rho = 0.8$. We have explained at the end of Subsubsec. 2.1.2 why we choose a small value of δ. Here and elsewhere we set $\delta = .1$. For the GPP(INID) case, we chose,

$$\hat{\gamma}_{n+1} = \frac{1}{n}\sum_{i=1}^{n} \gamma_i.$$

2. **DPP methods**

The DPP predictor is given in (20). The base measure F_0 was chosen to be the discrete uniform on $[\min\{y_1, \ldots, y_n\}, \max\{y_1, \ldots, y_n\}]$.
When we used a representative subset to form DPP_{rep} we used $K = 200$ as described at the end of Subsec. 7.1.

D.1.3. Hash function based methods

We used four hash function based predictors. Two were the mean and median of the EEDF generated by the Count-Min sketch. These are given in equations (68) and (69).

For the representative set forms of these two predictors, that we included only for the sake of comparison, as hash based methods are already one-pass, we followed the same procedure as for Shtarkov and Bayes.

D.1.4. Conformal

We simply used the technique of Bersson and Hoff (2024) who provided code. Their code was not one-pass so we only used their method with a representative subset. To be explicit, the conformity measure $C(y^{n+1})$ was the posterior from a normal density equipped with conjugate priors:

$$p(y_{n+1}|y^n) \quad = \quad \frac{\Gamma(\frac{2a_\sigma+1}{2})}{\sqrt{2a_\sigma\pi}\Gamma(\frac{2a_\sigma}{2})} \left(\frac{1}{\sqrt{\sigma_t^2}} \left(1 + \frac{1}{2a_\sigma} \frac{(y_i - \mu_\theta)^2}{\sigma_t^2}\right)^{-\frac{2a_\sigma+1}{2}} \right)$$

where

$$\tau_\theta^2 = \left(\frac{1}{\tau^2} + n\right)^{-1}, \mu_\sigma = \left(\frac{\mu}{\tau^2} + 1_n^T y^n\right)\tau_\theta^2, a_\sigma = a + n,$$
$$b_\sigma = b + y^{nT} y^n + \frac{\mu^2}{\tau^2} - (\tau_\theta^2)^{-1}\mu_\theta^2, \sigma_t^2 = \frac{b_\sigma}{a_\sigma}(1 + \tau_\theta^2).$$

For this method, the predictor is given by the mean of the prediction bounds from (70) with $\alpha = .15$, the default value in fabContinuousPrediction.

D.1.5. LSTM's

We used the keras package in R with its default settings unless otherwise noted here. As in Fig. 6, we used standard sigmoids and tanh's at the nodes. For a given length of streaming data, we used a burn-in of 10% of the data. When needed, we initialized parameter values randomly following the defaults. The optimization procedure was standard 'backprop through time' at each time step.

Loosely, in an LSTM model the individual LSTM's diagrammed in Fig. 6 can be used to farm a 'stack' of LSTM's (levels) or a sequence of LSTM's (layers) or both. This means that all the LSTM 'cells' used must be connected to the inputs and outputs. Describing this in detail is beyond our present scope.

Here, we simply used one layer of LSTM cells and either 20 or 50 LSTM units as levels; we only showed the results from 20 units because our representative sets were generally too small to make 50 units feasible.

The optimizer Adam optimizes over all parameters over all levels and the weights in the output step, a dense layer giving the final prediction.

An extra routine called dropout could be used to drop nodes randomly in the levels of the NN as a way to achieve robustness in estimating the weights. We did not do this because their concept of an SE doesn't exist in $\mathcal{M}$-open data, except as a measure of robustness.

D.2. Further examples

Here we provide graphs for four more data sets. The results from these graphs were used in Subsec. 7.3 and in particular in Table 3.

The four data sets are:

1. The Colombia data set can be found at https://data.world/hdx/f402d5ef-4a74-4036-8829-f04d6f38c8e9 which provides a description. It contains daily values of precipitation (mm) in Columbia over a period of four years ending in 2013. We used the first 5000 rows of the value column of the data set.
2. The Real Estate data set can be found at https://www.kaggle.com/datasets/derrekdevon/real-estate-sales-2001-2020. This data set contains information on real estate from all across the world. We used the first 10,000 rows and the column "Sales Ratio" which is the ratio of the Assessed value and the Sale Amount.
3. The Parking data set can be found at https://www.kaggle.com/datasets/mhmdkardosha/parking-birmingham. This data set contains information on parking capacity and parking occupancy of several car parks in Birmingham. We used the first 10000 rows and the column labeled "Occupancy".

The graphs of the sensitivity curves for all 14 methods follow.

For completeness we give a few details of our implementations.

- Except for the Colombia data set we used the same specifications for the other data sets as mentioned in the accelerometer data set. For the Colombia dataset we used $d_K = 10, K = 100$ for the hash function methods.
- For Real Estate, Fig. 10, $\sigma_{RV} = 630$ with τ at equal intervals of 90. The GPPRB(INID), GPPnoRB and the HBP median (rep.) methods had the best performances.
- For Parking, Fig. 11, $\sigma_{RV} = 198$ with τ located at equal intervals of 33. The HBP median (one pass) had the best performance.
- For Colombia, Fig. 12, $\sigma_{RV} = 900$ and τ is chosen at interval lengths of 100. The HBP median (one pass) and GPPRB methods had the best performance.

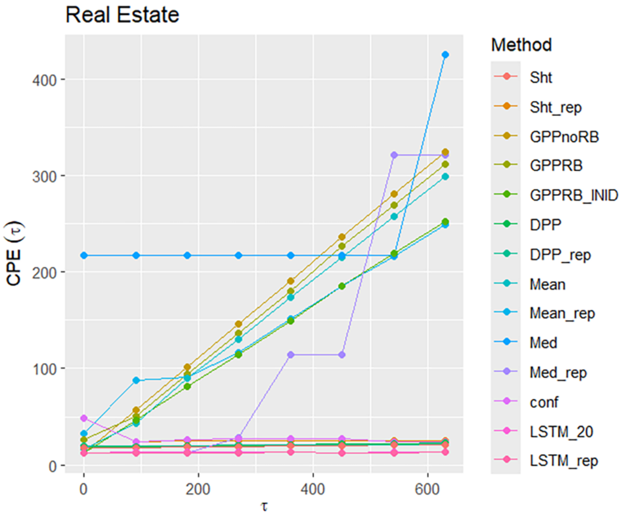

FIG 10. Real Estate *data.*

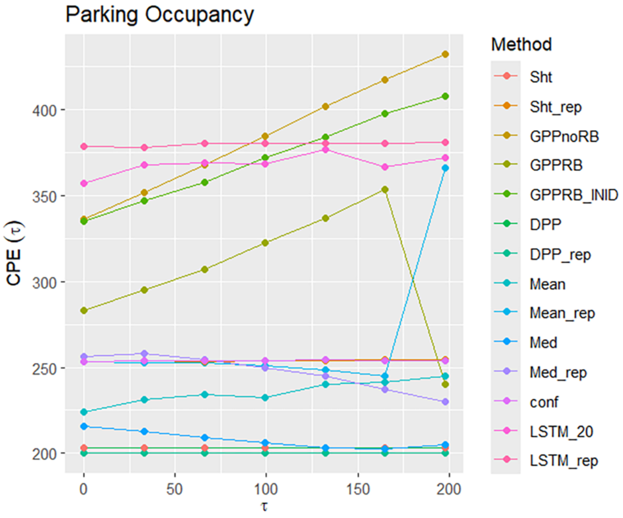

FIG 11. Parking *data.*

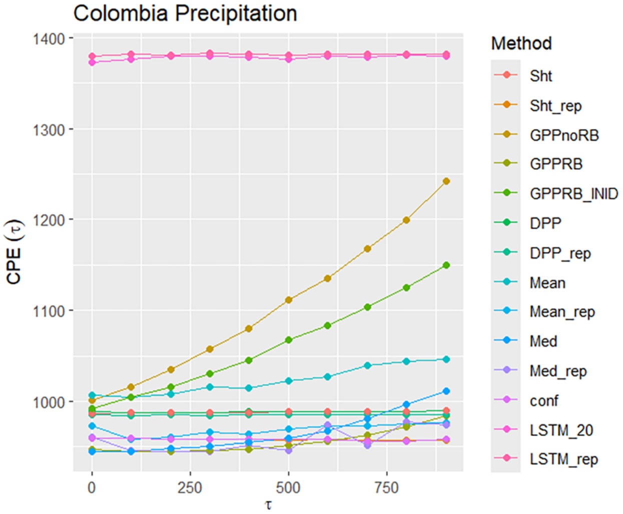

Fig 12. *Colombia precipitation.*

References

Charu C. Aggarwal. Recommender Systems. Springer International Publishing, Switzerland, 2016.

John Aitchison and Ian Dunsmore. Statistical Prediction Analysis. Cambridge University Press, Cambridge, 1975. ISBN 9780521206921. MR0408097

Mehmet Tahir Aslan. Customer shopping dataset. Kaggle, 2023. DOI:https://www.kaggle.com/datasets/mehmettahiraslan/customer-shopping-dataset.

Maroua Bahri, Silviu Maniu, and Albert Bifet. A sketch-based naive bayes algorithms for evolving data streams. In 2018 IEEE International Conference on Big Data, pages 604–613. IEEE, 2018.

Vineeth Balasubramanian, Shen-Shyang Ho, and Vladimir Vovk, editors. Conformal Prediction for Reliable Machine Learning: Theory, Adaptations and Applications. Elsevier, Amsterdam, 2014.

Andrew Barron, Jorma Rissanen, and Bin Yu. The minimum description length principle in coding and modeling. IEEE Transactions on Information Theory, 44(6):2743–2760, 1998. MR1658898

Andrew Barron, Teemu Roos, and Kazuho Watanabe. Bayesian properties of normalized maximum likelihood and its fast computation. In 2014 IEEE International Symposium on Information Theory, pages 1687–1691. IEEE, 2014. https://doi.org/10.1109/ISIT.2014.6875090.

Maurice Bartlett. The Statistical Analysis of Spatial Pattern. Chapman and hall, London, 1975. MR0402886

José M Bernardo and Adrian FM Smith. Bayesian Theory. John Wiley and Sons, Chichester, 2000. MR1274699

Elizabeth Bersson and Peter Hoff. Optimal conformal prediction for small areas. Journal of Survey Statistics and Methodology, 12(5):1464–1488, 2024.

Nicolo Cesa-Bianchi and Gábor Lugosi. Prediction, Learning, and Games. Cambridge University Press, 2006. MR2409394

Amit Chakrabarti. Data Stream Algorithms. Lecture notes from Dartmouth College, 2025.

Amit Chakrabarti, Andrew McGregor, and Anthony Wirth. Improved algorithms for maximum coverage in dynamic and random order streams. arXiv preprint arXiv:2403.14087, 2024. MR4802325

A. Chanda, N. V. Vinodchandran, and Bertrand Clarke. Point prediction for streaming data. In I. Kontoyiannis J. Klusowski and C. Rush, editors, Information Theory, Probability and Statistical Learning: A Festschrift in Honor of Andrew Barron. Springer, 2026.

Chan-Fu Chen. On asymptotic normality of limiting density functions with bayesian implications. Journal of the Royal Statistical Society Series B: Statistical Methodology, 47(3):540–546, 1985. MR0844485

Kai Lai Chung. A Course in Probability Theory. Academic Press, New York, 1974. ISBN 978-0-12-174650-6. MR0346858

B. Clarke and A.P Dawid. Online prediction with experts under a log-scoring rule. Unpublished manuscript, 1998.

Bertrand Clarke. Information optimality and Bayesian modelling. Journal of Econometrics, 138(2):405–429, 2007. MR2391317

Bertrand Clarke and Aleena Chanda. Online prediction for streaming observational data. arXiv preprint arXiv:2507.21308, 2025. MR4944388

Bertrand Clarke and Yuling Yao. A cheat sheet for Bayesian prediction. Statistical Science, 40(1):3–24, 2025. MR4859116

Graham Cormode and Shan Muthukrishnan. An improved data stream summary: the count-min sketch and its applications. Journal of Algorithms, 55(1):58–75, 2005. MR2132028

A Philip Dawid. The well-calibrated Bayesian. Journal of the American statistical Association, 77(379):605–610, 1982. MR0675887

A Philip Dawid. Present position and potential developments: Some personal views statistical theory the prequential approach. Journal of the Royal Statistical Society A, 147(2):278–290, 1984. MR0763811

Marcio Diniz, Rafael Izbicki, and Carlos Pereira. Bayesian and ridge regression models: a comparison by conformal prediction. Slides from a talk at the 2024 ISBA World Meeting.

R. M. Dudley. Uniform Central Limit Theorems. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 1999. MR1720712

Seymour Geisser. Predictive Inference. Chapman and Hall/CRC, New York, NY, 1993. MR1252174

Subhashis Ghosal. The Dirichlet process, related priors and posterior asymptotics. In Nils Lid Hjort, Chris Holmes, Peter Müller, and Stephen G. Walker, editors, Bayesian Nonparametrics, pages 35–79. Cambridge University Press, Cambridge, 2010. https://doi.org/10.1017/CBO9780511802478.003. MR2730660

Willian Oliveira Gibin. Sales in period walmart. Kaggle, 2023. https://www.kaggle.com/dsv/7013329.

Peter D Grünwald. The Minimum Description Length Principle. MIT press, Cambridge, MA, 2007.

Ramakrishna Gurajala, Praveen Choppala, James Meka, and Paul Teal. Derivation of the Kalman filter in a Bayesian filtering perspective. In 2nd International Conference on Range Technology, pages 1–5. IEEE, 2021.

Fares Hedayati and Peter Bartlett. The optimality of Jeffreys prior for online density estimation and the asymptotic normality of maximum likelihood estimators. In Proceedings of the 25th Annual Conference on Learning Theory, volume 23, pages 7.1–7.13, 2012.

So Hirai and Kenji Yamanishi. Efficient computation of normalized maximum likelihood codes for gaussian mixture models with its applications to clustering. IEEE Transactions on Information Theory, 59(11):7718–7727, 2013. MR3124670

Peter Hoff. Bayes-optimal prediction with frequentist coverage control. Bernoulli, 29(2):901–928, 2023. MR4550209

Ulf Johansson, Henrik Boström, Tuve Löfström, and Henrik Linusson. Regression conformal prediction with random forests. Machine learning, 97:155–176, 2014. MR3252831

Petri Kontkanen and Petri Myllymäki. A linear-time algorithm for computing the multinomial stochastic complexity. Information Processing Letters, 103(6):227–233, 2007. MR2339339

Wojciech Kotłowski and Peter Grünwald. Maximum likelihood vs. sequential normalized maximum likelihood in on-line density estimation. In Sham M. Kakade and Ulrike von Luxburg, editors, Proceedings of the 24th Annual Conference on Learning Theory, volume 19, pages 457–476. Proceedings of Machine Learning Research, 2011. URL https://proceedings.mlr.press/v19/kotlowski11a.html.

Tri Le and Bertrand Clarke. Using the Bayesian Shtarkov solution for predictions. Computational Statistics and Data Analysis, 104:183–196, 2016. https://doi.org/10.1016/j.csda.2016.06.018. URL https://doi.org/10.1016/j.csda.2016.06.018. MR3540994

Erich Lehmann. Testing Statistical Hypotheses. John Wiley and Sons, New York, NY, 1959. MR0107933

Erich Lehmann. Theory of Point Estimation. John Wiley and Sons, New York, NY, 1983. MR0702834

Jing Lei, James Robins, and Larry Wasserman. Distribution-free prediction sets. Journal of the American Statistical Association, 108(501):278–287, 2013. MR3174619

Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan Tibshirani, and Larry Wasserman. Distribution-Free Predictive Inference for Regression. Journal of the American Statistical Association, 113(523):1094–1111, 2018. MR3862342

Xiaohui Luo, Leonard Stefanski, and Dennis Boos. Tuning variable selection procedures by adding noise. Technometrics, 48(2):165–175, 2006. MR2277672

Steven MacEachern. Why not just use an additive bias? Personal communication, 2023.

Timo Mäkeläinen, Klaus Schmidt, and George Styan. On the existence and uniqueness of the maximum likelihood estimate of a vector-valued parameter in fixed-size samples. Annals of Statistics, pages 758–767, 1981. MR0619279

S. Muthukrishnan. Data streams: Algorithms and applications. Foundations and Trends Theoretical Computer Science, 1(2):117–236, 2005. https://doi.org/10.1561/0400000002. URL https://doi.org/10.1561/0400000002. MR2379507

S. Muthukrishnan. Data Stream Algorithms, 2009. Lecture notes from the 2009 Barbados Workshop on Computational Complexity, Bellairs Reseach Institute, Holetown, Barbados https://www.cs.mcgill.ca/~denis/notes09.pdf.

Bernt Øksendal. Stochastic Differential Equations: An Introduction with Applications. Springer, 6th edition, 2003. ISBN 978-3-642-34478-0. MR2001996

Giovanni Petris, Sonia Petrone, and Patrizia Campagnoli. Dynamic linear models. In Dynamic Linear Models with R, pages 31–84. Springer, 2009. MR2730074

David Ransohoff. Bias as a threat to the validity of cancer molecular-marker research. Nature Reviews Cancer, 5(2):142–149, 2005.

Francesco Ricci, Lior Rokach, and Bracha Shapira. Recommender systems: Techniques, applications, challenges. In Francesco Ricci, Lior Rokach, and Bracha Shapira, editors, Recommender Systems Handbook, pages 1–38. Springer, 2010. MR4454572

Jorma Rissanen. Universal coding, information, prediction, and estimation. IEEE Transactions on Information theory, 30(4):629–636, 1984. MR0755791

Jorma J Rissanen. Fisher information and stochastic complexity. IEEE Transactions on Information Theory, 42(1):40–47, 1996. MR1375327

Teemu Roos. Monte carlo estimation of minimax regret with an application to mdl model selection. In Proceedings of the 2008 IEEE Information Theory Workshop (ITW 2008), pages 284–288. IEEE, 2008. https://doi.org/10.1109/ITW.2008.4578670. URL https://doi.org/10.1109/ITW.2008.4578670.

Gustavo Sampaio, Arnaldo Filho, Leilton da Silva, and Leandro da Silva. Accelerometer. UCI Machine Learning Repository, 2019a. DOI: https://doi.org/10.24432/C5Q61V.

Gustavo Sampaio, Arnaldo Filho, Leilton da Silva, and Leandro da Silva. Prediction of motor failure time using an artificial neural network. Sensors, 19(19):4342, 2019b.

Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. Journal of Machine Learning Research, 9(3), 2008. MR2417240

Azeem Shaikh. The glivenko-cantelli theorem. URL: http://home.uchicago.edu/~amshaikh/webfiles/glivenko-cantelli.pdf, 2022.

Yurii Mikhailovich Shtarkov. Universal sequential coding of single messages. Problems of Information Transmission, 23(3):3–17, 1987. MR0914346

Atsushi Suzuki and Kenji Yamanishi. Exact calculation of normalized maximum likelihood code length using Fourier analysis. In 2018 IEEE International Symposium on Information Theory, pages 1211–1215, 2018.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. Conformal prediction: General case and regression. In Algorithmic Learning in a Random World, pages 19–69. Springer, 2022. MR2161220

Kenji Yamanishi. Learning with the Minimum Description Length Principle. Springer Nature, Singapore, 2023.