Probabilistic reconciliation of forecasts via importance sampling

Lorenzo Zambon*  Dario Azzimonti†  Giorgio Corani†

Abstract

Hierarchical time series are common in several applied fields. Forecasts are required to be coherent, that is, to satisfy the constraints given by the hierarchy. The most popular technique to enforce coherence is called reconciliation, which adjusts the base forecasts computed for each time series. However, recent works on probabilistic reconciliation present several limitations. In this paper, we propose a new approach based on conditioning to reconcile any type of forecast distribution. We then introduce a new algorithm, called Bottom-Up Importance Sampling, to efficiently sample from the reconciled distribution. It can be used for any base forecast distribution: discrete, continuous, or even in the form of samples. The method was tested on several temporal hierarchies showing that our reconciliation effectively improves the quality of probabilistic forecasts. Moreover, our algorithm is up to 3 orders of magnitude faster than vanilla MCMC methods.

Keywords: Probabilistic forecasting; Hierarchical time series; Reconciliation; Importance Sampling.

1 Introduction

Hierarchical time series are common in several fields, such as retail sales [Makridakis et al., 2021] or electricity demand [Taieb et al., 2021]. For instance, sales data may be aggregated into regions, or countries. Hierarchies may also be temporal: monthly time series may be aggregated to obtain quarterly or yearly time series, daily series may be aggregated into weekly series, and so on. Forecasts produced at different levels do not satisfy, in general, the constraints given by the hierarchy: for example, the sum of the monthly forecasts does not equal the yearly forecast. In this case, forecasts are said to be incoherent.
Reconciliation is performed to compute coherent forecasts from the base forecasts, which are typically incoherent. Besides producing coherent forecasts, reconciliation has been shown to improve the accuracy of the predictions [Kourentzes and Athanasopoulos 2021, Athanasopoulos et al. 2020]. Indeed, since the most disaggregated time series are typically very noisy, or even intermittent, they easily benefit from aggregated time series, which are usually smoother and thus less challenging to forecast.

Recent literature has mostly focused on point forecasts reconciliation [Hyndman et al. 2011, Di Fonzo and Marini 2011]. Reconciliation is particularly relevant for temporal hierarchies [Athanasopoulos et al. 2017, Kourentzes and Athanasopoulos 2019], in which time series are aggregated at different frequencies, such as monthly, quarterly and yearly. However, the whole forecast distribution is often needed for decision making or risk management, as point forecasts do not provide any uncertainty quantification. Probabilistic reconciliation has been studied in some recent works [Taieb et al. 2017, Jeon et al. 2019, Corani et al. 2020, Taieb et al. 2021]. A principled approach is proposed in Panagiotelis et al. [2022], where probabilistic reconciliation is formally defined as a projection. The parameters of the projection are optimized through Stochastic Gradient Descent in order to minimize some scoring rule. Besides being computationally intensive, this approach can not be used with discrete distributions. Corani et al. [2022] uses the soft evidence method to reconcile count time series. However, their implementation in probabilistic programming is computationally slow as it is based on MCMC.

In this paper, we provide a notion of coherence for probabilistic forecasts that is simple and rather general, as it can be applied to discrete and continuous distributions, and even non-linear constraints. We then propose a new approach to probabilistic reconciliation based on conditioning, which generalizes Corani et al. [2022]. It can be used with any type of base forecast distribution. Moreover, we propose the Bottom-Up Importance Sampling (BUIS) algorithm, a very efficient algorithm to sample from the reconciled distribution. A strength of BUIS is that it can be used even when the base forecast distribution is only available through samples. The numerical experiments on synthetic data show that BUIS outperforms MCMC in terms of computational time of 2 or 3 orders of magnitude. We test our method on temporal reconciliation of time series from real datasets, showing a significant improvement over base forecasts and over probabilistic reconciliation based on Gaussian assumptions Corani et al. [2020].

The paper is organized as follows. In Section 2 we set our notation and briefly recall point reconciliation. In Section 3 we obtain the expression of the reconciled distribution, and we compare our approach to the existing literature. The Bottom-Up Importance Sampling algorithm is introduced in Section 4. The numerical experiments are presented in Section 5 and Section 6, while conclusions and future work are in the last Section.
2 Notation

Consider a simple hierarchy, as in Figure 1. We denote by \( b = [b_1, \ldots, b_m]^T \) the vector of bottom observations, and by \( u = [u_1, \ldots, u_{n-m}]^T \) the vector of upper observations. We then denote by

\[
 y = \begin{bmatrix} u \\ b \end{bmatrix}
\]

the vector of all the observations. The hierarchy may be expressed as a set of linear constraints:

\[
 y = Sb, \quad \text{where} \quad S = \begin{bmatrix} A \\ I \end{bmatrix}.
\]

Here, \( I \in \mathbb{R}^{m \times m} \) is the identity matrix. \( S \in \mathbb{R}^{n \times m} \) is called summing matrix, while \( A \in \mathbb{R}^{(n-m) \times m} \) is called aggregating matrix. The constraint can thus be written as \( u = Ab \). For example, the aggregating matrix of the hierarchy in Figure 1 is given by

\[
 A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}.
\]

We then denote by \( S \) the set of coherent points, which is a linear subset of \( \mathbb{R}^n \):

\[
 S := \{ y \in \mathbb{R}^n : y = Sb \}.
\]

2.1 Temporal hierarchies

Temporal hierarchies [Athanasopoulos et al., 2017] are involved when forecasts are generated at different temporal scales. For instance, a quarterly time series may be aggregated to obtain semi-annual and annual series. If we are interested in predictions up to one year ahead, we compute the four quarterly forecasts \( \hat{q}_1, \hat{q}_2, \hat{q}_3, \hat{q}_4 \), the two semi-annual forecasts \( \hat{s}_1, \hat{s}_2 \), and the annual forecast \( \hat{a}_1 \). We then obtain the hierarchy in Figure 1 with
\( \mathbf{b} = [\hat{q}_1, \hat{q}_2, \hat{q}_3, \hat{q}_4] \) and \( \mathbf{u} = [\hat{a}_1, \hat{s}_1, \hat{s}_2] \). Forecasts computed at different frequencies are typically incoherent. For example, the quarterly predictions do not sum up to the annual prediction. Besides enforcing coherence, reconciliation has been shown to improve the accuracy of the predictions, also for the challenging case of intermittent time series [Kourentzes and Athanasopoulos 2021].

2.2 Point reconciliation

Let us now denote by \( \hat{\mathbf{y}} = \left[ \hat{\mathbf{u}} \mid \hat{\mathbf{b}} \right]^T \) the vector of base (incoherent) forecasts. Point reconciliation is typically performed in two steps. First, the reconciled bottom forecasts are computed by combining the base forecasts of the whole hierarchy:

\[
\tilde{\mathbf{b}} = \mathbf{G}\hat{\mathbf{y}},
\]

for some matrix \( \mathbf{G} \in \mathbb{R}^{m \times n} \). Then, the reconciled forecasts are obtained as

\[
\hat{\mathbf{y}} = \mathbf{S}\tilde{\mathbf{b}},
\]

where \( \mathbf{S} \) is the summing matrix. In this way, \( \hat{\mathbf{y}} \) is guaranteed to be coherent. For example, if we set \( \mathbf{G} = [\mathbf{I} \mid \mathbf{0}] \), we have the bottom-up approach, where the values of the upper series are completely ignored [Hyndman 2018, Chapter 11.2]. A popular choice for \( \mathbf{G} \) is provided by the minT method [Wickramasuriya et al. 2019]:

\[
\mathbf{G} = (\mathbf{S}^T\mathbf{W}^{-1}\mathbf{S})^{-1}\mathbf{S}^T\mathbf{W}^{-1},
\]

where \( \mathbf{W} \) is the covariance matrix of the errors of the base forecasts. This method has been proved to minimize the variance of the reconciled forecasts, under the assumption that the base forecasts are unbiased.

2.3 Probabilistic framework

In many applications, it is crucial for optimal decision making to take into account the uncertainty of the predictions [Gneiting and Katzfuss 2014]. In this case, we need to deal with models that are able to provide probabilistic forecasts, rather than point forecasts. In the probabilistic framework, forecasts are thus in the form of probability distributions. We denote by \( \hat{\nu} \in \mathcal{P}(\mathbb{R}^n) \) the forecast distribution for \( \mathbf{y} \), where \( \mathcal{P}(\mathbb{R}^n) \) is the space of probability measures on \( (\mathbb{R}^n, \mathcal{B}(\mathbb{R}^n)) \), and \( \mathcal{B}(\mathbb{R}^n) \) is the Borel \( \sigma \)-algebra on \( \mathbb{R}^n \). Moreover, we denote by \( \nu_u \) and \( \nu_b \) the marginal distributions of, respectively, the upper and the bottom components of \( \mathbf{y} \).

The forecast distribution \( \hat{\nu} \) may be either discrete or absolutely continuous. In the following, if there is no ambiguity, we will use \( \hat{\pi} \) to denote either its probability mass function, in the former case, or its density, in the latter. Therefore, if \( \hat{\nu} \) is discrete, we have

\[
\hat{\nu}(F) = \sum_{x \in F} \hat{\pi}(x),
\]

for any \( F \in \mathcal{B}(\mathbb{R}^n) \). Note that the sum is well-defined as \( \hat{\pi}(x) > 0 \) for at most countably many \( x \)’s. On the contrary, if \( \hat{\nu} \) is absolutely continuous,
for any $F \in \mathcal{B}(\mathbb{R}^n)$ we have
\[
\hat{\nu}(F) = \int_F \hat{\pi}(x) \, dx.
\]

## 3 Probabilistic Reconciliation

In this section, we recall the notion of coherence in the probabilistic framework, we describe our approach to probabilistic reconciliation, and we compare it with the existing literature.

Let $\hat{\nu} \in \mathcal{P}(\mathbb{R}^n)$ be a probabilistic forecast distribution. We say that $\hat{\nu}$ is incoherent if $\text{supp}(\hat{\nu}) \not\subseteq \mathcal{S}$. This means that there exists a set $T$ of incoherent points, i.e. $T \cap \mathcal{S} = \emptyset$, with $\hat{\nu}(T) > 0$. On the contrary, a coherent forecast distribution is supported on $\mathcal{S}$. Note that the map $s : \mathbb{R}^m \to \mathcal{S}$, defined as $s(b) = Sb$, is a measurable bijection, with inverse given by $s^{-1}(y) = b$, where $y = (u, b) \in \mathcal{S}$. Hence, as explained in Panagiotelis et al. [2022], for any $\nu \in \mathcal{P}(\mathbb{R}^m)$, we may obtain a coherent distribution as $\tilde{\nu} = s_#\nu$, namely the pushforward of $\nu$ using $s$:
\[
\tilde{\nu}(F) = \nu(s^{-1}(F)), \quad \forall F \in \mathcal{B}(\mathcal{S}),
\]
where $s^{-1}(F) := \{b \in \mathbb{R}^m : s(b) \in F\}$ is the pre-image of $F$.

In fact, it is easy to see that $s_#$ is a one-to-one correspondence between $\mathcal{P}(\mathbb{R}^m)$ and $\mathcal{P}(\mathcal{S})$. We thus propose to define as reconciled distribution any distribution $\nu \in \mathcal{P}(\mathbb{R}^m)$. It is a rather simple definition, and may be applied to any type of distribution. Moreover, it does not require $s$ to be a linear map, hence it could be applied to any problem where non-linear constraints are involved.

### 3.1 Probabilistic reconciliation

In the probabilistic framework, the aim of reconciliation is to obtain a reconciled distribution $\tilde{\nu}$ which is coherent from the base forecast distribution $\hat{\nu} \in \mathcal{P}(\mathbb{R}^n)$.

A naive approach could be to simply set $\tilde{\nu} = \nu_b$. In this way, all the information on the upper series is completely ignored. This method can be considered as the equivalent of the Bottom-Up approach for the probabilistic setting.

Panagiotelis et al. [2022] proposes a reconciliation method based on projections. Given a continuous map $\psi : \mathbb{R}^n \to \mathcal{S}$, the reconciled distribution is defined as $\tilde{\nu} = \psi_#\hat{\nu} \in \mathcal{P}(\mathcal{S})$, i.e. $\tilde{\nu}(F) = \hat{\nu}(\psi^{-1}(F))$, for any $F \in \mathcal{B}(\mathbb{R}^n)$. Since the map $\psi$ is expressed as $\psi = s \circ g$, with $g : \mathbb{R}^m \to \mathbb{R}^m$ and $s : \mathbb{R}^m \to \mathcal{S}$, the reconciled distribution may be equivalently defined as $\tilde{\nu} = g_#\hat{\nu} \in \mathcal{P}(\mathbb{R}^m)$. Note that, if $\hat{y}_1, \ldots, \hat{y}_N$ are independent samples from the base incoherent forecast distribution $\hat{\nu}$, then $\tilde{y}_1, \ldots, \tilde{y}_N$, defined as $\tilde{y}_i := g(\hat{y}_i)$ for $i = 1, \ldots, N$, are independent samples from the reconciled distribution $\tilde{\nu}$. The function $g : \mathbb{R}^n \to \mathbb{R}^m$ combines information
from all the levels by projecting on the bottom level the incoherent forecasts. For instance, if we define $g$ as $g(y) = b$, where $y = (u, b)$, we obtain $\tilde{\nu} = \nu_b$. In Panagiotelis et al. [2022], the map $g$ is assumed to be in the form $g(y) = d + Gy$, with $d \in \mathbb{R}^m$ and $G \in \mathbb{R}^{m \times n}$, and the parameter $\gamma := (d, \text{diag}(G))$ is optimized through stochastic gradient descent (SGD) to minimize the Energy Score [Szekely and Rizzo, 2013].

### 3.2 Probabilistic Reconciliation through conditioning

We now present our approach to probabilistic reconciliation, based on conditioning on the constraints given by the hierarchy. Let $Y = (U, B)$ be a random vector with law given by $\hat{\nu}$, so that $\nu_u$ and $\nu_b$ are the laws of, respectively, $U$ and $B$. Let us first suppose that the base forecast distribution $\hat{\nu} \in \mathcal{P}(\mathbb{R}^n)$ is discrete. We define $\tilde{\nu}$ by conditioning on the coherent subspace $S$:

$$\tilde{\nu}(F) = \frac{\mathbb{P}(B \in F \mid Y \in S)}{\mathbb{P}(Y \in S)} = \frac{\mathbb{P}(B \in F, U = AB)}{\mathbb{P}(U = AB)} = \frac{\sum_{b \in F} \hat{\pi}(Ab, b)}{\sum_{x \in \mathbb{R}^m} \hat{\pi}(Ax, x)}$$

(1)

for any $F \in \mathcal{B}(\mathbb{R}^m)$, provided that $\mathbb{P}(Y \in S) > 0$. Hence, $\tilde{\nu}$ is a discrete probability distribution with pmf given by

$$\tilde{\pi}(b) = \frac{\hat{\pi}(Ab, b)}{\sum_{x \in \mathbb{R}^m} \hat{\pi}(Ax, x)} \propto \hat{\pi}(Ab, b).$$

(2)

Let us now assume that $\hat{\nu}$ is absolutely continuous, and let $\tilde{\pi}$ denote its density. By analogy, the reconciled distribution $\tilde{\nu}$ should be given by

$$\tilde{\nu}(F) = \frac{\int_F \tilde{\pi}(Ab, b) \, db}{\int_{\mathbb{R}^m} \tilde{\pi}(Ax, x) \, dx},$$

(3)

so that $\tilde{\nu}$ is an absolutely continuous probability measure with density given by

$$\tilde{\pi}(b) \propto \hat{\pi}(Ab, b).$$

(4)

However, since the Lebesgue measure of $S$ is zero, we have that $\hat{\nu}(S) = 0$, hence $\mathbb{P}(B \in F \mid Y \in S)$ is not well-defined. To rigorously derive (3) and (4), we proceed as follows. Let us define the random vector $Z := U - AB$. Note that the event $\{Y \in S\}$ coincides with $\{Z = 0\}$. We can easily compute the joint density of $(Z, B)$ using the rule of change of variables [Billingsley, 2008, Chapter 17]:

$$\pi(Z, B)(z, b) = \hat{\pi}(z + Ab, b).$$


Then, the conditional density of $B$ given $Z = 0$ is given by [Cinlar, Chapter 4]:

$$
\tilde{\pi}(b) = \frac{\pi(Z, B)(0, b)}{\int_{\mathbb{R}^m} \pi(Z, B)(z, b) \, db} = \frac{\tilde{\pi}(Ab, b)}{\int_{\mathbb{R}^m} \tilde{\pi}(Ax, x) \, dx} \propto \hat{\pi}(Ab, b),
$$

provided that $\int_{\mathbb{R}^m} \tilde{\pi}(Ax, x) \, dx > 0$. Finally, note that, if $U$ and $B$ are independent, (2) and (4) may be rewritten as

$$
\tilde{\pi}(b) \propto \pi_u(Ab) \cdot \pi_b(b). \tag{5}
$$

From a Bayesian perspective, the derivation of the reconciled distribution described above may be interpreted in terms of a generalization of the Bayes’ rule. Indeed, the base distribution on the bottom variables may be interpreted as the prior:

$$
b \sim \nu_b, \tag{6}
$$

while the likelihood expresses the hierarchy constraints:

$$
\pi(u \mid b) = \delta_{u=Ab}. \tag{7}
$$

In this framework, the evidence is not given by a single observation, but rather by a probability distribution, i.e., the base conditional distribution of $U$ given $B$. In the area of Bayesian networks, this approach is known as updating using soft evidence [Darwiche, 2009, Chapter 3.6]. For more details, we refer to [Corani et al., 2022].

While in [Panagiotelis et al., 2022] the reconciled distribution was obtained by projecting the base distribution on $S$, in this work $\nu$ is obtained by conditioning $\hat{\nu}$ on the constraints given by the hierarchy. The behaviour of the base distribution outside the coherent subspace is ignored: intuitively, we do not take into account the probability of incoherent points, since they are not “admissible”. Indeed, it is clear from (4) and (5) that $\tilde{\nu}$ only depends on the values of $\hat{\nu}$ on $S$. The reconciled distribution satisfies the following property: for each pair of coherent points $y_1, y_2 \in S$, we have

$$
\frac{\tilde{\pi}(y_1)}{\tilde{\pi}(y_2)} = \frac{\pi(y_1)}{\pi(y_2)} \tag{8}
$$

if $\pi(y_2) \neq 0$, otherwise the reconciled is $\tilde{\pi}(y_2) = 0$. We say that reconciliation preserves the odds ratio: if, for example, $y_1$ is three times more likely than $y_2$ according to the base distribution, then it is the same also for the reconciled distribution. Moreover, our approach is rather general, since it is able to deal with both continuous and discrete distributions. Note that the approach based on projection involves, in most cases, to optimize the parameters through stochastic gradient descent. Besides being computationally expensive, it is not applicable for discrete distributions.
4 Sampling from the reconciled distribution

The reconciled distribution $\tilde{\nu}$ is not, in general, a known distribution. A significant exception is offered by the Gaussian case: if $\hat{\nu}$ is Gaussian, then $\tilde{\nu}$ is also Gaussian, and its mean and covariance matrix can be analytically computed; see [Corani et al., 2020] for a reconciliation method based on conditioning in the Gaussian case. In most cases, however, information about the distribution, such as the mean, variance, or quantiles, should be obtained through samples. Two of the most popular sampling techniques are Markov Chain Monte Carlo (MCMC, Metropolis et al., 1953) and Importance Sampling (IS, Kahn, 1950). MCMC is a broad family of algorithms, which consist in building a Markov Chain that has the target distribution as its stationary distribution. For a complete discussion on MCMC, we refer to Chib (2001). In the next subsection, we recall the main features of IS. For a complete discussion, see Tokdar and Kass (2010).

4.1 Importance Sampling

Let $X$ be an absolutely continuous random variable with density $p$. Suppose we want to compute an expectation $\mu = \mathbb{E}[f(X)]$, for some function $f$. If we are able to draw independent samples $x_1, \ldots, x_N$ from $p$, we can use the standard Monte Carlo estimate:

$$\mathbb{E}[f(X)] \approx \frac{1}{N} \sum_{i=1}^{N} f(x_i).$$  \hspace{1cm} (9)

In many cases, however, sampling from $p$ could be impractical, or it could lead to a very high variance of the Monte Carlo estimator (9). The idea of IS is to estimate the expectation $\mu$ by sampling from a different distribution $q$, and by weighting the samples to reflect the mismatch between the target $p$ and the proposal $q$. Let $q$ be a density such that $q(x) > 0$ if $f(x)p(x) \neq 0$, and let $y_1, \ldots, y_N$ be independent samples drawn from $q$. The importance sampling estimate is given by

$$\mathbb{E}[f(X)] \approx \frac{1}{N} \sum_{i=1}^{N} w(y_i) f(y_i),$$ \hspace{1cm} (10)

or by its self-normalized version

$$\mathbb{E}[f(X)] \approx \frac{\sum_{i=1}^{N} w(y_i) f(y_i)}{\sum_{i=1}^{N} w(y_i)},$$ \hspace{1cm} (11)

where $w$ is defined as $w(y) = \frac{q(y)p(y)}{q(y)}$. The function $w$ is often known only up to a normalizing constant: in this case, (11) must be used.

A common diagnostic to assess the efficiency of IS is the Effective Sample Size, which represents the number of independent samples from the target distribution that yields the same variance in the estimation. A
popular approximation of the ESS [Elvira et al., 2022] is given by

\[ \hat{ESS} = \left( \frac{\sum_{i=1}^{N} w(y_i)}{\sum_{i=1}^{N} w(y_i)^2} \right)^2. \] (12)

In the optimal case where \( q = p \), we have \( \hat{ESS} = N \). On the contrary, if the proposal \( q \) is not a good approximation of the target \( p \), we may expect that few weights are much larger than the others, leading to \( \hat{ESS} \approx 1 \).

Typically, the performance of IS drops dramatically [Tokdar and Kass, 2010] as the dimension of the space increases: this phenomenon is usually referred to as curse of dimensionality.

### 4.2 Probabilistic reconciliation via IS

Let \( \tilde{\nu} \), defined as in (1) for the discrete case or in (3) for the continuous case, be the target distribution. From now on, for the sake of simplicity, we will only refer to the continuous case, and we will denote by \( \tilde{\pi} \), defined as in (4), the density of \( \tilde{\nu} \). Notice, however, that everything is completely analogous even in the discrete setting, where \( \tilde{\pi} \), defined as in (2), denotes the probability mass function of \( \tilde{\nu} \).

We set \( \nu_b \), with density \( \pi_b \), as proposal distribution. Given a sample \( \mathbf{b}_1, \ldots, \mathbf{b}_N \) drawn from \( \nu_b \), the weights are computed as

\[ w_i := \frac{\tilde{\nu}(\mathbf{A} \mathbf{b}_i, \mathbf{b}_i)}{\pi_b(\mathbf{b}_i)}. \] (13)

Then, \( (\mathbf{b}_i, \tilde{w}_i)_{i=1}^{N} \) is a weighted sample from \( \tilde{\pi} \), where \( \tilde{w}_i := w_i / \sum_{j=1}^{N} w_j \) are the normalized weights. Note that (13) may be interpreted as the conditional density of \( U \) at the point \( \mathbf{A} \mathbf{b}_i \), given that \( B = \mathbf{b}_i \). Loosely speaking, we draw samples \( \mathbf{b}_i \)'s from the base distribution on the bottom nodes, and then weight how likely they are using the base distribution of the upper nodes. We thus combine the information contained in the distributions of both the bottom and the upper nodes. From a Bayesian perspective, we sample from the prior, and then we assign weights to the samples by using the soft evidence. Note that, under the assumption of independence between \( B \) and \( U \), the density of \( \tilde{\nu} \) may be factorized as in (5). In this case,

\[ w_i = \pi_u(\mathbf{A} \mathbf{b}_i). \] (14)

As mentioned before, the performance of IS is deeply influenced by the dimension of the space supporting the distributions and by the goodness of the choice of the proposal. Hence, we expect that the efficiency of IS drops as we deal with large hierarchies. We also expect a similar behavior if the incoherence is large, that is, when the probability of \( \mathbf{A} U - B \) being close to zero is low. Indeed, in this case, the proposal distribution \( \nu_b \) is not a good approximation of the target distribution \( \tilde{\nu} \). In Appendix A, we run some experiments that confirm this behaviour. This justifies the introduction of a new algorithm, called Bottom-Up Importance Sampling, that is more robust with respect to the hierarchy size and the incoherence level.
4.3 Bottom-Up Importance Sampling algorithm

First, we make two assumptions:

**Assumption 1.** The data structure is hierarchical.

**Assumption 2.** The base forecasts of each variable are conditionally independent, given the time series observations.

Assumption 1 means that the data structure disaggregates in a unique hierarchical manner [Hyndman et al., 2008]. Hence, it is represented as a tree, in which every node only has one parent. Assumption 2 is the main limit of our algorithm. Nevertheless, we believe it is a reasonable assumption in many contexts, such as temporal hierarchies or count data time series, in which forecasts are typically produced independently for each time series.

Under Assumption 1 and Assumption 2, we develop a new algorithm, called Bottom-Up Importance Sampling. The core idea is to split a single \((n-m)\)-dimensional Importance Sampling problem into \(n-m\) one-dimensional problems. To do so, we start by drawing a sample from the base distribution \(\nu_b\). Then, for each level of the hierarchy, from bottom to top, we update the sample through an Importance Sampling step. At each step, the “partially” reconciled distribution is used as proposal. In this way, we encapsulate the information contained in the base distributions as \(l\) steps, hence \(l\) times.

In the following, \(\pi\) denotes either the density or the probability mass function, depending on whether we deal with continuous or discrete distributions. The BUIS algorithm is reported below (Algorithm 1).

The “Resample” step is performed by sampling with replacement from the discrete distribution given by

\[
\mathbb{P} \left( \mathbf{b} = \left( b_{1,1}^{(i)}, \ldots, b_{q,l,i}^{(i)} \right) \right) = w^{(i)},
\]  

for all \(i = 1, \ldots, N\). We explicit below the BUIS algorithm on the simple hierarchy in Figure 1.

1. Sample \((b_j^{(i)})_{i=1,\ldots,N}\) from \(\pi_{B_j}\), for \(j = 1, 2, 3, 4\)
2. Compute the weights \((w^{(i)})_{i=1,\ldots,N}\) with respect to \(U_2\) as

\[
w^{(i)} = \pi_{U_2} \left( b_1^{(i)} + b_2^{(i)} \right)
\]
Algorithm 1 Bottom-Up Importance Sampling

1: Sample \((b^{(i)})_i=1,\ldots,N\) from \(\pi_b\)
2: for \(l\) in levels do
3:   for \(j = 1, \ldots, k_l\) do
4:     \(\hat{w}^{(i)} \leftarrow \pi_{u_j,i}(\sum_{t=1}^{q_{j,l}} b_{t,(j,l)}^{(i)})\) for \(i = 1, \ldots, N\)
5:     \(w^{(i)} \leftarrow \frac{\hat{w}^{(i)}}{\sum_h \hat{w}^{(h)}}\) for \(i = 1, \ldots, N\)
6:     \((\bar{b}_j^{(i)})_i \leftarrow \text{Resample} \left(\left(b_{1,(j,l)}^{(i)}, \ldots, b_{q_{j,l},(j,l)}^{(i)}\right), \left(w^{(i)}\right)_i\right)\)
7:   end for
8: \(b^{(i)} \leftarrow [\bar{b}_1^{(i)}, \ldots, \bar{b}_{k_l}^{(i)}]\) for \(i = 1, \ldots, N\)
9: end for
10: return \((b^{(i)})_i\)

3. Sample \((\tilde{b}_1^{(i)}, \tilde{b}_2^{(i)})_i\) with replacement from \((b_1^{(i)}, b_2^{(i)}, w^{(i)})_i=1,\ldots,N\)
4. Repeat step 2 and 3 using \(B_3, B_4\) and \(U_3\) to get \((\tilde{b}_3^{(i)}, \tilde{b}_4^{(i)})_i\)
5. Set \(\left(b_1^{(i)}, b_2^{(i)}, b_3^{(i)}, b_4^{(i)}\right)_i = \left(\tilde{b}_1^{(i)}, \tilde{b}_2^{(i)}, \tilde{b}_3^{(i)}, \tilde{b}_4^{(i)}\right)_i\) and move to the next level
6. Compute the weights \((w^{(i)})_i=1,\ldots,N\) with respect to \(U_1\) as
\[w^{(i)} = \pi_{U_1} \left(\hat{b}_1^{(i)} + b_2^{(i)} + b_3^{(i)} + b_4^{(i)}\right)\]
7. Sample \((\tilde{b}_1^{(i)}, \tilde{b}_2^{(i)}, \tilde{b}_3^{(i)}, \tilde{b}_4^{(i)})_i\) with replacement from \((b_1^{(i)}, b_2^{(i)}, b_3^{(i)}, b_4^{(i)}, w^{(i)})_i\)

In Appendix B, we prove that the output of the BUIS is approximately a sample drawn from the reconciled distribution \(\tilde{\nu}\).

4.4 Sample-based BUIS

The densities of the forecast distributions are not always available in analytical form. Indeed, several prediction methods provide probabilistic forecasts only in the form of samples. This means that we are not able to analytically evaluate the density at any given point. However, we are able to perform reconciliation even without the analytical form of the densities. Since we only deal with one-dimensional densities to compute the weights, we may effectively use approximations based on samples. For discrete distributions, the most reasonable choice is to simply use the empirical density. As for the continuous setting, several methods are available to approximate the true density, such as kernel density estimation [Chen, 2017].

Therefore, we only need to replace line 4 in Algorithm 1 with:
4.5 Grouped time series

In some cases, time series have a data structure that does not disaggregate in a unique hierarchical manner. They are usually referred to as grouped time series [Hyndman, 2018, Chapter 11]. For instance, consider a weekly time series, for which we compute the following temporal aggregates: 2-weeks, 4-weeks, 13-weeks, 26-weeks, 1-year. If we deal with one year forecasts, we have 52 bottom time series and $26 + 13 + 4 + 2 + 1 = 46$ upper time series. Clearly, this structure cannot be represented as a tree. Since Assumption 1 is not satisfied, the BUIS algorithm cannot be used. Indeed, as highlighted in the proof, we need $\overline{b}_1, \ldots, \overline{b}_k$ to be independent to be allowed to multiply the densities. This is not true if Assumption 1 is not in force, since correlations between bottom nodes are created when conditioning on the upper levels.

However, when dealing with grouped time series, one may proceed as follows. First, find the largest sub-hierarchy within the group structure. For instance, in the example above, we consider the sub-hierarchy given by the bottom time series and by the 2-weeks, 4-weeks and 1-year aggregates. All the other upper nodes are then regarded as additional constraints. We use the BUIS algorithm on the sub-hierarchy, obtaining a sample $b$. Then, we compute the weights on $b$ using the distributions given by the additional constraints. This is equivalent to performing a plain IS, where we use the output of BUIS on the hierarchical part as proposal distribution. In this way, we reduce the dimension of the IS task from $n - m$, the total number of upper constraints, to the number of constraints that are not included in the sub-hierarchy: in the above example, from 46 to 6. We highlight that the distribution we sample from would be the same even with different choices of sub-hierarchies. However, picking the largest one is of course the best choice from a computational perspective.

5 Experiments on synthetic data

We test our algorithm on synthetic data. We use two different hierarchies: a simple binary tree, which is a proper hierarchy, and thus satisfies Assumption 1, and a weekly hierarchy, as described in Section 4.5. For each node, we fix the parameters of the base distribution. We run the experiments in both the continuous and discrete setting, using, respectively, Gaussian and Poisson distributions. We draw samples from the reconciled distribution using Importance Sampling, Bottom-Up Importance Sampling and Markov Chain Monte Carlo. For the latter, we use the Python library PyMC3 [Salvatier et al., 2016], which implements an adaptive Metropolis-Hastings algorithm [Haario et al., 2001] for discrete distributions and the No-U-Turn Sampler (NUTS) [Hoffman et al.].
In the Gaussian case, we are able to compare our results to the exact solution, which is analytically computable [Corani et al., 2020]. For the Poisson case, our reference method is MCMC.

5.1 Binary hierarchy

We set a simple binary hierarchy with 8 bottom variables and 7 upper variables, as in Figure 2.

5.1.1 Gaussian distributed forecasts

On each bottom node, we set a Gaussian distribution with mean randomly chosen between 5 and 10, and standard deviation $\sigma_b = 3$. We denote by $m_b \in \mathbb{R}^8_+$ the vector of the base bottom means. For each upper node, we set a Gaussian distribution with standard deviation $\sigma_u = 3$. We set the base means of the upper variables, for each incoherence level $\epsilon = 0.1, 0.3, 0.5, 0.8$, as $m_u = (1 + \epsilon)A m_b$, where $A$ is the aggregating matrix. Hence, an incoherence level of 0.5 means that the base upper means are 50% greater than the sum of the corresponding base bottom means. We also set $\sigma_u = 1$.

We perform reconciliation, for each incoherence level, using IS and BUIS with 100,000 samples, and with MCMC using 4 chains with 5000 samples each. We repeat the experiments 30 times.

For each method and for each node we compute the mean of the reconciled distribution by taking the sample mean. We compare it to the true mean of the reconciled distribution, analytically computed, using the mean absolute percentage error (MAPE, Hyndman [2006]). It is defined as

$$\text{MAPE}(\bar{y}, \hat{y}) = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{\bar{y}_i - \hat{y}_i}{\bar{y}_i} \right| \cdot 100,$$

where $\bar{y}$ is the reconciled mean and $\hat{y}$ is the true mean. MAPEs are reported in Table 1. As the incoherence level grows, the error obtained using IS grows to over 12%. The performance of BUIS, instead, is comparable to that of MCMC in terms of MAPE. Even with a very large incoherence, the error using BUIS is about 0.3%. 

![Figure 2: A binary hierarchy.](image)
Table 1: Mean Absolute Percentage Error on the reconciled mean, using binary hierarchy and Gaussian distributions.

| $\epsilon$ | IS     | BUIS   | MCMC   |
|------------|--------|--------|--------|
| 0.1        | 0.17%  | 0.11%  | 0.12%  |
| 0.3        | 0.33%  | 0.11%  | 0.10%  |
| 0.5        | 1.75%  | 0.13%  | 0.08%  |
| 0.8        | 12.39% | 0.31%  | 0.07%  |

Figure 3: Boxplot of the reconciled mean of a bottom variable, using binary hierarchy and Gaussian distributions. We compare IS, BUIS and MCMC for all incoherence levels. The blue line corresponds to the exact value.

In Figure 3 we show the boxplot of the reconciled mean, over the 30 iterations, of a bottom node, for each method and incoherence level. The blue line corresponds to the exact value. The standard deviation using IS, as $\epsilon$ grows, gets very large compared to BUIS and MCMC. In order to have a clear comparison between BUIS and MCMC, in Figure 4 are shown the same boxplots with only BUIS and MCMC. Only for the extreme value $\epsilon = 0.8$, MCMC appears to perform better than BUIS.

We also compare the shape of the distributions obtained with IS, BUIS and MCMC. To do so, on each bottom node, we compute the 2-Wasserstein distance [Panaretos and Zemel, 2019] between the true reconciled distribution and the empirical distribution obtained with the three different methods. The average distances are reported in Table 2. As a comparison, we also include the $W_2$ distance between the reconciled distribution and the base, unreconciled distribution. The results show that BUIS clearly outperforms IS, and that, except for the extreme value $\epsilon = 0.8$, it is comparable to MCMC.
Figure 4: Boxplot of the reconciled mean of a bottom variable, using binary hierarchy and Gaussian distributions. We only compare BUIS and MCMC. The blue line corresponds to the exact value.

| $\epsilon$ | IS   | BUIS | MCMC | base |
|------------|------|------|------|------|
| 0.1        | 0.041| 0.028| 0.031| 0.950|
| 0.3        | 0.094| 0.031| 0.030| 2.155|
| 0.5        | 0.521| 0.042| 0.031| 3.483|
| 0.8        | 2.973| 0.116| 0.032| 5.511|

Table 2: Average Wasserstein distance between the reconciled distribution on the bottom nodes obtained with different methods and the true reconciled distribution, in the case of binary hierarchy and Gaussian distributions.

|          | IS   | BUIS | MCMC |
|----------|------|------|------|
| Average time | 0.06 s | 0.17 s | 33.9 s |

Table 3: Average computational time for binary hierarchy and Gaussian distributions.

Finally, the average computational times of the three different methods are reported in Table 3. Note that BUIS is about 200 times faster than MCMC. This significant computational gain is explained by the fact that, using IS, both sampling and computing the weights are done simultaneously for all the samples. When using MCMC, on the other hand, sampling and computing the acceptance probability must be done sequen-
tially. For more details, see Appendix C.

5.1.2 Poisson distributed forecasts

Now, we set a Poisson distribution on each bottom variable, with mean
randomly chosen in the interval $[5, 10]$. We denote by $\lambda_b \in \mathbb{R}_+^n$ the vector
of the base bottom means. As before, for each incoherence level $\epsilon = 0.1, 0.3, 0.5, 0.8$, we set the mean of the upper variables as $\lambda_u = (1 + \epsilon)A\lambda_b$.

We perform reconciliation using Importance Sampling and Bottom-Up Importance Sampling with 100,000 samples, and with MCMC using 4 chains with 5,000 samples each. We also use the BUIS algorithm without the analytical probability mass function, but only using samples, as described in Section 4.4. We repeat all experiments 30 times.

For each method, we compute the mean of the mean of the reconciled
distribution at each node. We compare the results obtained using IS, BUIS, and BUIS using samples with the results obtained using MCMC. The Mean Absolute Percentage Errors are reported in Table 4.

| $\epsilon$ | IS   | BUIS  | BUIS w/samples |
|------------|------|-------|---------------|
| 0.1        | 0.36%| 0.37% | 0.37%         |
| 0.3        | 0.35%| 0.32% | 0.33%         |
| 0.5        | 0.51%| 0.33% | 0.33%         |
| 0.8        | 3.82%| 0.37% | 0.37%         |

*Table 4: Mean Absolute Percentage Error on the reconciled mean, using binary hierarchy and Poisson distributions. The reference method is MCMC.*

As before, in Figure 5 is reported the boxplot of the reconciled mean of
a bottom node over the 30 iterations. In order to have a clear comparison
between BUIS and MCMC, IS is not shown. The results obtained using
BUIS, in both cases, are similar to those obtained using MCMC.

Finally, the average computational times are reported in Table 5. As in
the continuous case, BUIS is much faster than MCMC. When we only use
samples, the algorithm is slower, but is still about 25 faster than MCMC.

| Average time |
|--------------|
| IS | BUIS | BUIS w/samples | MCMC |
| 0.12 s | 0.22 s | 1.42 s | 35.5 s |

*Table 5: Average computational time for binary hierarchy and Poisson distributions.*
Figure 5: Boxplot of the reconciled mean of a bottom variable, using binary hierarchy and Poisson distributions. We compare BUIS, BUIS with samples and MCMC for all incoherence levels.

5.2 Weekly hierarchy

We repeat similar experiments using a weekly hierarchy with 52 bottom nodes and 46 upper nodes. We run the experiments in both the Gaussian and Poisson cases, using different incoherence levels: $\epsilon = 0.1, 0.3, 0.5$. Since this is not a proper hierarchy, i.e., assumption H1 is not in force, in order to use BUIS we need to proceed as described in Section 4.5. We only compare BUIS and MCMC, since the dimension of the space is too large to use IS. In the Poisson case, we also run BUIS with samples.

The Mean Absolute Percentage Errors in the Gaussian setting are reported in Table 6, while the MAPEs with respect to our reference method (MCMC) in the Poisson settings are reported in Table 7. Even in the case of such a large hierarchy, using BUIS we are able to achieve a very small error. Finally, the average computational times are reported in Table 8. Note that BUIS is about 3 orders of magnitude faster than MCMC.

| $\epsilon$ | BUIS   | MCMC   |
|------------|--------|--------|
| 0.1        | 0.07 % | 0.06 % |
| 0.3        | 0.09 % | 0.05 % |
| 0.5        | 0.21 % | 0.04 % |

Table 6: Mean Absolute Percentage Error on the reconciled mean, using weekly hierarchy and Gaussian distributions.

Table 7: Mean Absolute Percentage Error on the reconciled mean, using weekly hierarchy and Poisson distributions.

Table 8: Average computational times in seconds.
Table 7: Mean Absolute Percentage Error on the reconciled mean, using weekly hierarchy and Poisson distributions. The reference method is MCMC.

| ε   | BUIS | BUIS w/samples |
|-----|------|----------------|
| 0.1 | 0.34 % | 0.33 % |
| 0.3 | 0.36 % | 0.36 % |
| 0.5 | 1.09 % | 1.07 % |

Table 8: Average computational time for weekly hierarchy.

|          | BUIS | BUIS w/samples | MCMC     |
|----------|------|----------------|----------|
| Gaussian | 1.47 s | -              | 1065.4 s |
| Poisson  | 2.25 s | 9.99 s         | 2417.8 s |

6 Experiments

We now perform probabilistic reconciliation on temporal hierarchies, using time series extracted from two different data sets: carparts, available from the R package `expsmooth` [Hyndman, 2015], and syph, available from the R package ZIM [Yang et al., 2018]. The carparts data set is about monthly sales of car parts. As in [Hyndman et al., 2008, Chapter 16], we remove time series with missing values, with less than 10 positive monthly demands and with no positive demand in the first 15 and final 15 months. After this selection, there are 1046 time series left. Monthly data are aggregated into 2-months, 3-months, 4-months, 6-months and 1-year levels. The syph data set is about the weekly number of syphilis cases in the United States. We remove the time series with ADI greater than 20. The ADI is computed as $\text{ADI} = \sum_{i=1}^{P} p_i$, where $p_i$ is the time period between non-zeros values and $P$ is the total number of periods [Syntetos and Boylan, 2005]. We also remove the time series corresponding to the total number of cases in the US. After this selection, there are 50 time series left. Weekly data are aggregated into 2-weeks, 4-weeks, 13-weeks, 26-weeks and 1-year levels. In both cases, they are grouped time series, as described in Section 4.5.

For both data sets, we fit a Generalized Linear Models using the tscount package [Liboschik et al., 2017]. We use a negative binomial predictive distribution, with a first-order regression on past observations. The test set has length 1 year for both data sets. We thus compute up to 12 steps ahead at monthly level, and up to 52 steps ahead at weekly level.
Probabilistic forecasts are returned in the form of samples.

Reconciliation is performed in three different ways. In the first case, we fit a Gaussian distribution on the returned samples. Then, we follow Corani et al. [2020] to analytically compute the Gaussian reconciled distribution. In the second case, we fit a negative binomial distribution on the samples, and we reconcile using the BUIS algorithm. Finally, we use the BUIS directly on the samples, as explained in Section 4.4. This method yields a computational gain, since fitting a distribution on the samples may be computationally expensive. We refer to these methods, respectively, as $N$, $NB$, and $samples$. Furthermore, we denote by $base$ the unreconciled forecasts.

We use different indicators to assess the performance of each method. The mean scaled absolute error (MASE) [Hyndman, 2006] is defined as

\[
    MASE = \frac{\text{MAE}}{Q},
\]

where \( \text{MAE} = \frac{1}{h} \sum_{h=1}^{T} |y_{t+j} - \hat{y}_{t+j}| \) and \( Q = \frac{1}{T-1} \sum_{t=2}^{T} |y_{t} - y_{t-1}| \). Here, \( y_{t} \) denotes the value of the time series at time \( t \), while \( \hat{y}_{t+j} \) denotes the point forecast computed at time \( t \) for time \( t + h \). The median of the distribution is used as point forecast, since it minimizes MASE [Kolassa, 2016].

The mean interval score (MIS) [Gneiting, 2011] is defined, for any \( \alpha \in (0, 1) \), as

\[
    MIS = (u - l) + \frac{2}{\alpha} (l - y) \mathbb{1}(y < l) + \frac{2}{\alpha} (y - u) \mathbb{1}(y > u),
\]

where \( l \) and \( u \) are the lower and upper bounds of the \((1 - \alpha)\) forecast coverage interval and \( y \) is the value of the time series. In the following, we use \( \alpha = 0.1 \). MIS penalizes wide prediction intervals, as well as intervals that do not contain the true value.

The rank probability score (RPS) [Kolassa, 2016] is defined as the \( l^2 \)-distance between the predictive cumulative distribution function (CDF) and the true CDF, i.e. the step function at the true value of the time series.

Finally, the Enery score [Székely and Rizzo, 2013] is defined as

\[
    ES(P, y) = \mathbb{E}_P \left[ \| y - s \|^\alpha \right] - \frac{1}{2} \mathbb{E}_P \left[ \| s - s' \|^\alpha \right],
\]

where \( P \) is the forecast distribution on the whole hierarchy, \( s, s' \sim P \) a pair of independent random variables and \( y \) the vector of the actual values of all the time series. The energy score is a proper scoring rule for distributions defined on the entire hierarchy [Panagiotelis et al., 2022]. We compute \( ES \), with \( \alpha = 2 \), using samples, as explained in Wickramasuriya [2021].

We use the skill score to compare the performance of a method with respect to a baseline method, in terms of percentage improvement. For example, if we take $base$ as a baseline, the skill score of $NB$ on MASE is given by

\[
    \text{Skill}(NB, base) = \frac{\text{MASE}(base) - \text{MASE}(NB)}{(\text{MASE}(base) + \text{MASE}(NB))/2}.
\]
The skill scores for carparts are reported in Table 9. We only compute MASE, MIS, and ES, since RPS is computationally expensive. In the first two columns, we compare the performance of the reconciled forecasts NB and samples with N. Both NB and samples methods yield a significant improvement for all the indicators, and for all the hierarchy levels. In the last two columns, we compare NB and samples with the base forecasts. For both methods, the average improvement is over 18% for MASE, about 40% for MIS and over 50% for ES.

The skill scores for syph are reported in Table 10. In the first two columns are reported the performance of NB and samples with respect to N; as before, the average improvement is significant for all indicators. Finally, the last two columns show an average improvement of NB and samples over base of more than 8% for MASE, more than 6% for MIS, more than 18% for RPS and more than 11% for ES. On average, we observe that samples performs better than NB.

Table 9: Skill score results on the time series extracted from carparts, detailed by each level of the hierarchy.

| metric | hier-level | NB vs N | samples vs N | NB vs base | samples vs base |
|--------|------------|---------|--------------|------------|----------------|
| ES     |            | 0.459   | 0.469        | 0.518      | 0.527          |
| MASE   | Monthly    | 1.124   | 1.112        | 0.140      | 0.129          |
|        | 2-Monthly  | 0.678   | 0.678        | 0.250      | 0.271          |
|        | Quarterly  | 0.504   | 0.529        | 0.209      | 0.264          |
|        | 4-Monthly  | 0.458   | 0.462        | 0.159      | 0.207          |
|        | Biannual   | 0.375   | 0.365        | 0.140      | 0.157          |
|        | Annual     | 0.378   | 0.365        | 0.179      | 0.174          |
|        | average    | 0.586   | 0.585        | 0.180      | 0.200          |
| MIS    | Monthly    | 0.500   | 0.665        | 0.449      | 0.631          |
|        | 2-Monthly  | 0.172   | 0.325        | 0.445      | 0.559          |
|        | Quarterly  | 0.217   | 0.297        | 0.433      | 0.465          |
|        | 4-Monthly  | 0.300   | 0.338        | 0.349      | 0.356          |
|        | Biannual   | 0.389   | 0.314        | 0.367      | 0.264          |
|        | Annual     | 0.494   | 0.331        | 0.396      | 0.225          |
|        | average    | 0.346   | 0.378        | 0.406      | 0.417          |
\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|c|}
\hline
\textbf{metric} & \textbf{hierc-level} & $NB \text{ vs } N$ & $\text{samples vs } N$ & $NB \text{ vs } \text{base}$ & $\text{samples vs } \text{base}$ \\
\hline
\textbf{ES} & & 0.076 & 0.119 & 0.114 & 0.151 \\
\hline
\textbf{MASE} & Weekly & 0.732 & 0.731 & 0.141 & 0.140 \\
 & 2-Weekly & 0.461 & 0.460 & 0.156 & 0.139 \\
 & 4-Weekly & 0.266 & 0.265 & 0.128 & 0.120 \\
 & Quarterly & 0.065 & 0.090 & 0.011 & 0.037 \\
 & Biannual & 0.052 & 0.106 & 0.066 & 0.152 \\
 & Annual & 0.027 & 0.067 & -0.005 & 0.037 \\
 & \textit{average} & 0.267 & 0.287 & 0.083 & 0.104 \\
\hline
\textbf{MIS} & Weekly & 0.453 & 0.443 & 0.460 & 0.451 \\
 & 2-Weekly & 0.250 & 0.246 & 0.327 & 0.335 \\
 & 4-Weekly & 0.159 & 0.217 & 0.189 & 0.251 \\
 & Quarterly & 0.014 & 0.061 & -0.111 & -0.082 \\
 & Biannual & -0.010 & 0.048 & -0.271 & -0.211 \\
 & Annual & 0.038 & 0.059 & -0.234 & -0.223 \\
 & \textit{average} & 0.151 & 0.179 & 0.060 & 0.087 \\
\hline
\textbf{RPS} & Weekly & 0.642 & 0.625 & 0.535 & 0.469 \\
 & 2-Weekly & 0.421 & 0.413 & 0.370 & 0.339 \\
 & 4-Weekly & 0.248 & 0.258 & 0.216 & 0.222 \\
 & Quarterly & 0.045 & 0.077 & 0.006 & 0.050 \\
 & Biannual & 0.045 & 0.096 & -0.024 & 0.054 \\
 & Annual & 0.037 & 0.078 & -0.009 & 0.038 \\
 & \textit{average} & 0.240 & 0.258 & 0.182 & 0.195 \\
\hline
\end{tabular}

\caption{Skill score results on the time series extracted from syph, detailed by each level of the hierarchy.}
\end{table}

7 Conclusions

In this paper, we propose a new approach to probabilistic reconciliation based on conditioning, rather than projecting as in [Panagiotelis et al. 2022]. Our approach can be applied to any type of base forecast distribution. We also propose the BUIS algorithm, which is able to efficiently sample from the reconciled distribution. This algorithm can be used even if the base distributions are only available in the form of samples, which is the case for several commonly used forecasting methods. The numerical experiments on temporal hierarchies show a significant improvement of our method compared to the base forecasts. We believe that the main
limit of our algorithm is the assumption of conditional independence of
the forecast distributions. Future research directions thus include deal-
ing with correlations between the base forecasts, as well as probabilistic
cross-temporal reconciliation.

References

George Athanasopoulos, Rob J Hyndman, Nikolaos Kourentzes, and Fo-
tios Petropoulos. Forecasting with temporal hierarchies. *European
Journal of Operational Research*, 262(1):60–74, 2017.

George Athanasopoulos, Puwasala Gamakumara, Anastasios Panagi-
telis, Rob J Hyndman, and Mohamed Affan. Hierarchical forecast-
ing. In *Macroeconomic forecasting in the era of big data*, pages 689–719.
Springer, 2020.

Patrick Billingsley. *Probability and measure*. John Wiley & Sons, 2008.

Yen-Chi Chen. A tutorial on kernel density estimation and recent ad-
vancess. *Biostatistics & Epidemiology*, 1(1):161–187, 2017.

Siddhartha Chib. Chapter 57 - markov chain monte carlo meth-
ods: Computation and inference. volume 5 of *Handbook of Econo-
metrics*, pages 3569–3649. Elsevier, 2001. doi: https://doi.org/10.
1016/S1573-4412(01)05010-3. URL https://www.sciencedirect.com/
science/article/pii/S1573441201050103.

Erhan Çinlar. *Probability and stochastics*, volume 261. Springer.

Giorgio Corani, Dario Azzimonti, Joao PSC Augusto, and Marco Zaffalon.
Probabilistic reconciliation of hierarchical forecast via Bayes’ rule. In *Proc. European Conf. On Machine Learning and Knowledge Discovery
in Database ECML/PKDD*, volume 3, pages 211–226, 2020.

Giorgio Corani, Nicolo Rubattu, Dario Azzimonti, and Alessandro An-
tonucci. Probabilistic reconciliation of count time series. *arXiv preprint
arXiv:2207.09322*, 2022.

Adnan Darwiche. *Modeling and reasoning with Bayesian networks*. Cam-
bridge university press, 2009.

Tommaso Di Fonzo and Marco Marini. Simultaneous and two-step recon-
ciliation of systems of time series: methodological and practical issues.
*Journal of the Royal Statistical Society: Series C (Applied Statistics)*,
60(2):143–164, 2011.

Víctor Elvira, Luca Martino, and Christian P. Robert. Rethinking the
effective sample size. *International Statistical Review*, 2022.

Tilmann Gneiting. Quantiles as optimal point forecasts. *International
Journal of forecasting*, 27(2):197–207, 2011.
Tilmann Gneiting and Matthias Katzfuss. Probabilistic forecasting. Annual Review of Statistics and Its Application, 1:125–151, 2014.

Heikki Haario, Eero Saksman, and Johanna Tamminen. An adaptive metropolis algorithm. Bernoulli, pages 223–242, 2001.

Matthew D Hoffman, Andrew Gelman, et al. The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. J. Mach. Learn. Res., 15(1):1593–1623, 2014.

G. Hyndman, R.J. & Athanasopoulos. Forecasting: principles and practice, 2nd edition, OTexts: Melbourne, Australia, 2018. URL OTexts.com/fpp2.

Rob Hyndman. Another look at forecast-accuracy metrics for intermittent demand. Foresight: The International Journal of Applied Forecasting, 4(4):43–46, 2006.

Rob Hyndman, Anne B Koehler, J Keith Ord, and Ralph D Snyder. Forecasting with exponential smoothing: the state space approach. Springer Science & Business Media, 2008.

Rob J Hyndman. expsmooth: Data Sets from "Forecasting with Exponential Smoothing", 2015. URL https://CRAN.R-project.org/package=expsmooth R package version 2.3.

Rob J. Hyndman, Roman A. Ahmed, George Athanasopoulos, and Han Lin Shang. Optimal combination forecasts for hierarchical time series. Computational Statistics & Data Analysis, 55(9):2579 – 2589, 2011. ISSN 0167–9473.

Jooyoung Jeon, Anastasios Panagiotelis, and Fotios Petropoulos. Probabilistic forecast reconciliation with applications to wind power and electric load. European Journal of Operational Research, 279(2):364–379, 2019. doi: https://doi.org/10.1016/j.ejor.2019.05.020.

Herman Kahn. Random sampling (monte carlo) techniques in neutron attenuation problems. i. Nucleonics (US) Ceased publication, 6(See also NSA 3-990), 1950.

Stephan Kolassa. Evaluating predictive count data distributions in retail sales forecasting. International Journal of Forecasting, 32(3):788–803, 2016.

Nikolaos Kourentzes and George Athanasopoulos. Cross-temporal coherent forecasts for australian tourism. Annals of Tourism Research, 75: 393–409, 2019.

Nikolaos Kourentzes and George Athanasopoulos. Elucidate structure in intermittent demand series. European Journal of Operational Research, 288(1):141–152, 2021.
Tobias Liboschik, Konstantinos Fokianos, and Roland Fried. tscount: An R package for analysis of count time series following generalized linear models. *Journal of Statistical Software*, 82(5):1–51, 2017.

Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. The m5 competition: Background, organization, and implementation. *International Journal of Forecasting*, 2021.

Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

Anastasios Panagiotelis, Puwasala Gamakumara, George Athanasopoulos, and Rob J Hyndman. Probabilistic forecast reconciliation: Properties, evaluation and score optimisation. *European Journal of Operational Research*, 2022.

Victor M Panaretos and Yoav Zemel. Statistical aspects of wasserstein distances. *Annual review of statistics and its application*, 6:405–431, 2019.

Db Rubin and Db Rubin. Using the sir algorithm to simulate posterior distributions. 1988.

John Salvatier, Thomas V Wiecki, and Christopher Fonnesbeck. Probabilistic programming in Python using PyMC3. *PeerJ Computer Science*, 2:e55, 2016.

A. F. M. Smith and A. E. Gelfand. Bayesian statistics without tears: A sampling-resampling perspective. *The American Statistician*, 46(2):84–88, 1992. ISSN 00031305. URL http://www.jstor.org/stable/2684170

Aris A. Syntetos and John E. Boylan. The accuracy of intermittent demand estimates. *International Journal of Forecasting*, 21(2):303–314, 2005. ISSN 0169-2070.

Gábor J Székely and Maria L Rizzo. Energy statistics: A class of statistics based on distances. *Journal of statistical planning and inference*, 143 (8):1249–1272, 2013.

Souhaib Ben Taieb, James W. Taylor, and Rob J Hyndman. Coherent probabilistic forecasts for hierarchical time series. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 3348–3357, 2017.

Souhaib Ben Taieb, James W Taylor, and Rob J Hyndman. Hierarchical probabilistic forecasting of electricity demand with smart meter data. *Journal of the American Statistical Association*, 116(533):27–43, 2021.

Surya T Tokdar and Robert E Kass. Importance sampling: a review. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(1):54–60, 2010.
Shanika L. Wickramasuriya. Probabilistic forecast reconciliation under the gaussian framework. *arXiv preprint arXiv:2103.11128*, 2021.

Shanika L. Wickramasuriya, George Athanasopoulos, and Rob J. Hyndman. Optimal forecast reconciliation for hierarchical and grouped time series through trace minimization. *Journal of the American Statistical Association*, 114(526):804–819, 2019.

Ming Yang, Gideon Zamba, and Joseph Cavanaugh. *ZIM: Zero-Inflated Models (ZIM) for Count Time Series with Excess Zeros*, 2018. URL https://CRAN.R-project.org/package=ZIM. R package version 1.1.0.
A ESS of IS for reconciliation

We run some experiments to test the dependence of the efficiency of IS from the size of the hierarchy and from the incoherence level. We set a binary hierarchy, described by a tree where each node has 1 parent and 2 children (except for the top and the bottom nodes). If \( k \) is the number of levels of the hierarchy, then there are \( m = 2^k \) bottom nodes and \( 1 + 2 + \cdots + 2^{k-1} \) upper nodes. An example for \( k = 3 \) levels is reported in Figure 6.

The base distribution is defined as follows. We set an independent Poisson distribution on each node. For all bottom nodes, we arbitrarily set the means \( \lambda_b \in \mathbb{R}^m \). Then, we fix an incoherence level \( \epsilon > 0 \), and we generate the means of the upper nodes as \( \lambda_u = (1 + \epsilon)A\lambda_b \).

We set \( \epsilon = 0.2 \), and for each hierarchy size \( k \in \{2, 3, 4, 5\} \) we draw 100,000 weighted samples from \( \tilde{\nu} \) and we compute the effective sample size. We repeat 30 times and take the average. The results are reported in Figure 7a. Then, we set \( k = 3 \) and we do the same for \( \epsilon \in \{0.1, 0.2, \ldots, 0.8\} \). The results are reported in Figure 7b.

As expected, the effective sample size dramatically drops as the hierarchy size or the incoherence level grows. Note that, in Figure 7, the y axis is logarithmic. The Bottom-Up Importance Sampling algorithm is proposed to overcome this issue.
B Proof of BUIS algorithm

We show that the output \((\hat{b}^{(i)})\) of the BUIS algorithm is approximately a sample drawn from the target distribution \(\hat{\nu}\).

From [5], and from Assumption 2 we have that

\[
\hat{\pi}(b) \propto \pi_u(b) \cdot \pi_u(AB) = \prod_{t=1}^m \pi_{b_t}(b_t) \cdot \prod_{i=1}^L \prod_{j=1}^{k_i} \pi_{u_{j,l},i} \left( \sum_{k=1}^{q_{j,l}} b_{k,(j,l)} \right),
\]

where we are using the notation of Section 4.3. The initial distribution of the sample \((\hat{b}^{(i)})\) is given by \(\pi_b = \prod_{t=1}^m \pi_{b_t}(b_t)\). We show that each iteration of the algorithm corresponds to multiplying by a \(\pi_{u_{j,l}} \left( \sum_{k=1}^{q_{j,l}} b_{k,(j,l)} \right)\) term.

Let \(\pi_X\) be a density over \(\mathbb{R}^d\), and \(w : \mathbb{R}^d \to \mathbb{R}\) a continuous function. Let \(X_1, \ldots, X_N\) be independent samples from \(\pi_X\), and compute the unnormalized weights \((\hat{w}^{(i)})_{i=1,\ldots,N}\) as \(\hat{w}^{(i)} = w(X_i)\). Then, if we draw \(Y_1, \ldots, Y_n\) from the discrete distribution given by

\[
P(Y = X_i) = \hat{w}^{(i)}, \quad i = 1, \ldots, N,
\]

where \(\hat{w}^{(i)} = \frac{\hat{w}^{(i)}(X_i)}{\sum_{q_{j,l}=1}^{q_{j,l}} \hat{w}^{(i)}(X_{i^q})}\), then \((Y_i)_{i=1,\ldots,n}\) is approximately an IID sample from the density \(\pi_X(x) \propto \pi_Y(x) \cdot w(x)\). This technique is known as importance resampling [Rubin and Rubin 1988] or weighted bootstrap [Smith and Gelfand 1992]. The same holds also for discrete distributions, using the pmf instead of the density.

Hence, if we compute the weights \(w^{(i)}\)'s as in the algorithm and sample \((\hat{b}^{(i)}(i))\) from \([15]\), it is approximately equivalent to sampling from \(\pi_u(b) \cdot \pi_{u_{j,l},i} \left( \sum_{k=1}^{q_{j,l}} b_{k,(j,l)} \right)\), where \(\pi_u\) is the original density of \((b_{1,(j,l)}, \ldots, b_{q_{j,l},(j,l)})\).

In other words, the weighting-resampling step corresponds to multiplying the density of the sample by a \(\pi_{u_{j,l}} \left( \sum_{k=1}^{q_{j,l}} b_{k,(j,l)} \right)\) term.

Finally, note that in this way we are conditioning with respect to \(u_{j,l}\).

After the weighting-resampling step, \((b_{1,(j,l)}, \ldots, b_{q_{j,l},(j,l)})\) are correlated. Since, from [11] the hierarchy is given by a tree, we are guaranteed that for any level \(l\) and for all \(j = 1, \ldots, k_l\), \(\hat{b}_j\) only depends on \(b_{1,(j,l)}, \ldots, b_{q_{j,l},(j,l)}\), \(u_{j,l}\) and each upper variable that is under \(u_{j,l}\). From [12] we have that \(\hat{b}_1, \ldots, \hat{b}_{k_l}\) are independent. Hence, the density of \([\hat{b}_1, \ldots, \hat{b}_{k_l}]\) is given by the product of the densities of all \(\hat{b}_j\)'s, and the proof is concluded.

C Comparison between MCMC and IS

In order to fully understand the reasons for the significant difference in computational time between the MCMC and the IS approach, we compare the two methods on a minimal example. Let us consider a hierarchy given by two bottom observations, \(b_1\) and \(b_2\), and just one upper observation \(u\),
which is the sum of $b_1$ and $b_2$. We set a Gaussian distribution for each observation.

We implement a simple Metropolis-Hastings algorithm with a Gaussian proposal distribution with fixed variance $\tau I$ to sample from the reconciled distribution $\tilde{\pi}(b) = \pi_{b_1}(b_1) \cdot \pi_{b_2}(b_2) \cdot \pi_u(b_1 + b_2)$. The algorithm reads as follows:

**Initialize $b^{(0)}$**

for $j = 1, \ldots, N$ do

- Sample $y^{(j)} \sim N(b^{(j-1)}_1, \tau I)$
- $\alpha \leftarrow \min \left(1, \frac{\tilde{\pi}(y^{(j)})}{\pi_u(b^{(j-1)})} \right)$
- $u \leftarrow \text{Unif}(0,1)$
- if $u < \alpha$ then
  - $b^{(j)} \leftarrow y^{(j)}$
- else
  - $b^{(j)} \leftarrow b^{(j-1)}$

end if

end for

return ($b^{(i)}$, $w_i$)

On a standard laptop, it takes about 4 seconds to get a 10,000 samples from $\tilde{\pi}$. In particular, most of the time is employed by the computation of the acceptance probability $\alpha$, which requires about $3.7 \cdot 10^{-4}$ seconds per loop. Sampling from the proposal distribution only requires about $3 \cdot 10^{-5}$ seconds.

We then implement an IS algorithm on the same hierarchy:

**Sample $b^{(1)}, \ldots, b^{(N)} \text{ IID } \pi_\theta**

$w_i \leftarrow \pi_u \left( \hat{b}_1^{(i)} + \hat{b}_2^{(i)} \right)$

return ($b^{(i)}$, $w_i$)

It takes about $7 \cdot 10^{-3}$ seconds to draw 100,000 IID samples from $\pi_\theta$, and about the same time to compute all the weights. The significant improvement in computational time using IS instead of MCMC is due to the fact that both sampling and computation of the weights are done simultaneously for all the samples, rather than sequentially as in MCMC.