Gaussian Process Models for Incremental Loss Ratios

Mike Ludkovski ludkovski@pstat.ucsb.edu and Howard Zail hzail@elucidor.com

August 4, 2020

Abstract

We develop Gaussian process (GP) models for incremental loss ratios in loss development triangles. Our approach brings a machine-learning, spatial-based perspective to stochastic loss modeling. GP regression offers a non-parametric probabilistic distribution regarding future losses, capturing uncertainty quantification across three distinct layers: model risk; correlation risk; extrinsic uncertainty due to randomness in observed losses. To handle statistical features of loss development analysis, namely spatial non-stationarity, convergence to ultimate claims, and heteroskedasticity, we develop several novel implementations of fully-Bayesian GP models. We perform extensive empirical analyses over the NAIC loss development database across six business lines, comparing and demonstrating the strong performance of our models. Our computational work is performed using R and Stan programming environments and is publicly shareable.

1 Introduction

In P&C reserving practice, estimation of reserves is a learning task: how to use the history of paid claims to predict the pattern of emergence of future claims. A major ongoing challenge is to quantify the respective uncertainty or distribution of future claims emergence. This is important for determining the required reserves, risk capital and the resultant solvency of an insurance company.

In this paper, we explore the use of Gaussian Process (GP) models for loss development. GP regression is a powerful machine learning technique for empirical model building that has become a centerpiece of the modern data science toolkit. GP approaches now form an extensive ecosystem and are fast popularized across a wide range of predictive analytics applications. We posit that GPs are also highly relevant in the context of risk analytics by offering both full uncertainty quantification and a hierarchical, data-driven representation of nonlinear models. The proposed GP framework for loss triangles revolves around viewing the emergence of claims by accident and development year as a random field, bringing to the forefront a spatial statistics perspective. More precisely, we provide a fully-Bayesian GP approach for modeling of claims development that yields a full probabilistic distribution regarding future losses. We then show how GPs can be adapted to deal with the considerable complexities of loss development distributions including non-stationarity, heteroscedasticity and domain constraints.

1.1 Models for Loss Development

The loss development challenge is to project from the "upper triangle" of historic claims, the distribution of the lower triangle of future claims. The mean of such distribution is typically used to determine the level of loss reserves, and the tail of the distribution is used to set the desired level of allocated risk capital. The fundamental objects available in loss development are cumulative paid claims, "CC", indexed in terms of Accident Year p (`AY`), and Development Lag q (`DL`). In the NAIC dataset used below (see Section 1.5) we have $CC_{p,q}$ for $p = 1988, \ldots, 1997$, $q = 1, \ldots, 10 = Q$, where Q denotes the full development horizon (10 years in NAIC data). Two derived quantities are the incremental payments

$$I_{p,q} := \begin{cases} CC_{p,1} & q = 1; \\ CC_{p,q} - CC_{p,q-1}, & q = 2, \ldots, Q, \end{cases} \tag{1.1}$$

and the Year-over-Year loss development factors (LDF), i.e. the consecutive ratios between cumulative paid claims,

$$F_{p,q} := \frac{CC_{p,q}}{CC_{p,q-1}}, \qquad q = 2, \ldots. \tag{1.2}$$

The advantage of LDFs is that they are normalized in terms of overall claims volume. A related metric to incremental claims is the Incremental Loss Ratio (ILR),

$$L_{p,q} := I_{p,q}/P_p, \tag{1.3}$$

where P_p is the premium for the p-th Accident Year. LDF and ILR are referred to as "Modeling Factors" in this paper and denoted by Y.

The training or historic data is presented via a triangle of development claims. We view the triangle as a collection of cells, indexed by two "inputs", namely (i) AY^i, and (ii) DL^i, and one "output", for example L^i. We are interested in linking inputs $x^i = [AY^i, DL^i]$, to the outputs $y^i = L^i$ (or $y^i = F^i$), converting the loss development triangle/square into a columnar set of data with a single index i. The paradigm we adopt is a *response surface approach*, i.e. construction of a statistical latent surface representing the "true" output, which is then noisily observed in actual data. Thus, we postulate that there is an underlying function $x \mapsto f(x)$ such that

$$y^i = f(x^i) + \epsilon^i. \tag{1.4}$$

Following the triangular structure we henceforth index cells by their Year and Lag p, q, writing AY_p, DL_q. For concreteness, we focus on the ILRs $L_{p,q} \equiv Y_{p,q}$ as the outputs; then $\{f_{p,q} \equiv f(x)\}$ is a true incremental loss development structure that describes the completed square, and ϵ^i are the observation noises, which are independent, with state-dependent variance $\mathrm{Var}(\epsilon_{p,q}) = \sigma^2_{p,q}$. For example, the Chain Ladder (CL) method assumes that the LDF $F_{p,q}$ is log-normal with a mean μ_q (i.e. independent of Accident Year), prescribed via the CL formula, and variance dependent on $CC_{p,q}$. More generally, we construct a statistical model relating the observed factors Y's to the true latent input-output relationship, plus observation noise, $y^i \sim \mathcal{L}(f(x^i))$ where $\mathcal{L}$ captures the likelihood of realized observations.

In this paradigm, forecasting future losses translates into the double step of *inference* on the latent $\{f_{p,q}\}$ and then constructing the *predictive distribution* of future modeling factors $Y_{p,q}$'s that is based on inferring the structure of $\{\epsilon_{p,q}\}$. For instance in the CL, the inferred logarithmic loss development factors μ_q are added together to inflate to the ultimate log development factor, combined with the (additive on the log scale) noises $\epsilon_{p,q}$ and finally exponentiated to obtain $CC_{p,Q}$.

1.2 Uncertainty Quantification

Our primary goal is to extrapolate existing losses to complete the full loss square given experience up to the present date YR. This extrapolation must recognize that loss development is subject to stochastic shocks and hence the forecasted $Y_{p,q}, p+q > \text{YR}$ should be represented as random variables. Indeed, beyond providing the *expected* modeling factor $\mathbb{E}[Y_{p,q}]$ (or more commonly expected cumulative ultimate losses $\mathbb{E}[CC_{p,Q}]$), we wish to quantify the respective distribution, in order for example to assess the quantile of $CC_{p,Q}$ as needed in solvency computation. Since we have the fundamental relationship that

$$CC_{p,Q} = \begin{cases} CC_{p,q} \prod_{\ell=q}^{Q-1} \frac{CC_{p,\ell+1}}{CC_{p,\ell}} = CC_{p,q} \prod_{\ell=q}^{Q-1} F_{p,\ell} & \text{or} \\ CC_{p,q} + P_p \left(\sum_{\ell=q+1}^{Q} L_{p,\ell} \right) \end{cases} \tag{1.5}$$

to forecast unpaid claims we need to forecast future $L_{p,\ell}$'s or $F_{p,\ell}$'s.

The overall task above of constructing a predictive distribution of $CC_{p,q}$ conditional on losses so far introduces the fundamental distinction between extrinsic and intrinsic uncertainty. The extrinsic uncertainty, also known as process variance, is linked to the fact that in (1.4) observed losses are noisy, so one must first forecast the latent surface $f_{p,q}$ and then overlay the extrinsic noise $\epsilon_{p,q}$. Thus, extrinsic uncertainty is the variance $\text{Var}(\epsilon_{p,q}) = \sigma^2_{p,q}$. In contrast, intrinsic or model or parameter uncertainty refers to the impossibility of fully learning the true data generating process $f_{p,q}$ given a finite amount of loss data. Traditionally, extrinsic uncertainty is fitted through a parametric assumption (e.g. log-normal LDFs or Poisson incremental claims), while intrinsic uncertainty is obtained from the standard error of a point estimate $f_{p,q}$, inferred through Maximum Likelihood Estimation or via ad hoc formulas such as the CL.

Many of existing models tend to under-estimate intrinsic uncertainty which leads to under-predicting the variance of ultimate claims. As a result, the models do not statistically validate in large-scale out-of-sample tests for reserving risk, due to being light-tailed. Thus actual ultimate claims are frequently too much above or too much below the mean forecast, relative to the estimated predictive variance. One popular remedy has been Bayesian frameworks (England et al., 2012; Kuo, 2019; Taylor, 2015; Wüthrich, 2018; Zhang and Dukic, 2013; Zhang et al., 2012) that more explicitly control the residual randomness in $\{f_{p,q}\}$ through the respective priors. In this article, we contribute to this strand by

developing a machine-learning-inspired, hierarchical approach that includes *three layers* of uncertainty.

First, we adopt a semi-parametric extrinsic uncertainty, prescribing the distributional family of $\epsilon_{p,q}$, but allowing for cell-dependence. As our main example, we take $\epsilon_{p,q}$ to be Normally distributed, but with variance σ_q^2 depending on development year.

Second, we model $f_{p,q}$ in terms of a Gaussian random field. This offers a natural quantification of model risk through the two-step process of conditioning $f_{p,q}$ on the observed loss triangle and then computing the residual posterior variance in $f_{p,q}$. The latter posterior variance summarizes our confidence in correctly inferring $f_{p,q}$ given the data and is referred to as intrinsic uncertainty.

Third, we account for the misspecification risk regarding the spatial structure of f by a further Bayesian Markov Chain Monte Carlo (MCMC) procedure. Specifically, we employ MCMC to learn the distribution of the *hyperparameters*, especially the lengthscales, of the Gaussian process representing f. This is the correlation uncertainty layer. We show that in aggregate these three uncertainty layers not only properly account for overall predictive uncertainty (which is no longer under-estimated), but also generate non-parametric predictive distributions while maintaining computational tractability afforded by the Gaussian structure.

To summarize, in our GP-based framework, completing the triangle is comprised of (i) sampling, via MCMC, hyperparameters ϑ of f consistent with observations $\mathcal{D}$; (ii) sampling a realization of $f_{p,q}$'s conditional on ϑ and $\mathcal{D}$; (iii) sampling a realization of future observation noise $\epsilon_{p,q}$; (iv) combining $f_{p,q}$'s and $\epsilon_{p,q}$'s to return the (sample-based) distribution of $CC_{p,Q}$ via (1.5). Importantly, while observation noises ϵ are a priori independent across cells, the spatial dependence in $f_{p,q}$ creates correlation in Modeling Factors across both accident years and development lags. This is a major distinction from traditional methods where no such correlation is available. Moreover, the GP paradigm allows to produce directly the whole vector $\{f_{p,q+1}, f_{p,q+2}, \ldots, f_{p,Q}\}$ that gives a full projection of the claims between today and the ultimate development date.

1.3 Related literature

Stochastic loss reserving addresses uncertainty quantification of unpaid non-life insurance losses. The vast literature is centered around the Chain Ladder (CL) method (Mack, 1993, 1994) that provides analytic predictive distributions based on empirical plug-in estimates, and Bayesian methods (England and Verrall, 2002) that extract posterior distributions through bootstrapping or Markov chain Monte Carlo (MCMC) approaches. We refer to Wüthrich and Merz (2008) and Meyers (2015) monographs for an overview of the state-of-the-art.

The CL approach provides empirical formulas for the latent LDFs $\{f_{p,q}\}$ which are then commonly injected into a log-normal model for $F_{p,q}$. Of note, the CL mean μ_q can be interpreted as a weighted average of $F_{\cdot,q}$ across accident years, $f_q = \sum_\ell w_{\ell,q} F_{\ell,q}$. To provide statistical underpinnings to the CL formulas, Bayesian CL frameworks have been

developed, see Merz and Wüthrich (2010). The latter derive μ_q as the posterior mean of a certain Bayesian conditioning expression. In turn, this allows the natural extension to consider intrinsic uncertainty, i.e. to use the full posterior of μ_q to capture model risk. Another notable extension of CL considers correlation in $\epsilon_{p,q}$ across accident years to remedy the respective independence assumption in the original CL which does not validate well. Such correlation is also a popular way to boost the variance of ultimate claims. A related issue is including a payment year trend.

In the multivariate Gaussian reserving models (Wüthrich and Merz, 2008, Ch. 4), the cumulative claims $CC_{p,q}$ are modeled as a log-normal random variable with mean/standard deviations $\mu_{p,q}, \sigma_{p,q}$. The typical assumption is that the logged mean is of the form $\mu_{p,q} = \alpha_p + \beta_q$, decomposing the true surface as an additive combination of 1-dimensional Accident Year $\boldsymbol{\alpha}$ and Development Lag $\boldsymbol{\beta}$ trend factors (an idea similar to the Age-Period decomposition of mortality surfaces). The variances $\sigma^2_{p,q}$ are then directly specified via a correlation matrix Σ. In the most common version they are taken to be independent across cells; see for instance Meyers (2015) who proposed the Levelled CL (LCL) approach where the observation variance σ^2_q is constant across accident years and decreases in Lag q. The three latent vectors $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\sigma}$ are estimated via MCMC (implemented in Stan), after specifying certain structured priors on them. In the alternative correlated chain ladder (CCL) proposal Meyers (2015), the mean structure is augmented with $\mu_{p,q} = \alpha_p + \beta_q + \rho(\log CC_{p-1,q} - \mu_{p-1,q})$ generating across-year correlation between cumulative paid claims. However, in his results the posterior of the estimated correlation coefficient ρ is quite wide, being both positive and negative, precluding making a clear conclusion regarding the across-year dependence. Another parametrization was done in Zhang and Dukic (2013); Zhang et al. (2012) who forecast cumulative losses with a prescribed nonlinear functional form, fitted through a hierarchical Bayesian model. Statistically similar is the approach by England and Verrall (2002) who used generalized additive models based on cubic splines; see also the extension to a GAMLSS model in Spedicato et al. (2014).

An alternative is to directly model $CC_{p,q}$. This paradigm dispenses with the existence of a "true" data-generating loss process and directly views triangle completion as an extrapolation task: given $\{CC_{p,q}, p+q \leq YR\}$ we seek to build an extrapolating surface that matches the observed upper triangle and smoothly extends it into the bottom half. In this framework, there is no observation noise, but the surface is viewed as a random field that intrinsically "vibrates". Projected losses are then obtained by conditioning on historical observations but otherwise taking the remaining intrinsic variability in $F_{p,q}$ as the measure of the randomness of future LDFs. This approach will lead to correlated fluctuations in $F_{p,q}$ due to the spatial covariance.

This idea was recently implemented by Lally and Hartman (2018) (henceforth LH) who proposed to apply GP regression to run-off triangles. In their framework the conditional residual variance of $f_{p,q}$ is precisely the predictive uncertainty. This idea exploits the fact that $\text{Var}(f_{p,q}) \to \eta^2$ for far-out extrapolation. LH construct GP regression models for $CC_{p,q}$ viewing cumulative losses as a smooth function of accident period and development

lag. The inherent spatial nonstationarity of the run-off triangle is handled through input warping that is learned as part of fitting the GP model. LH work in a Bayesian framework, capturing hyperparameter uncertainty, as well as uncertainty about the best warping function.

1.4 Contributions

Our proposal to employ GP models for loss developments combines the existing concept of a Bayesian Chain Ladder with a spatial representation of loss emergence. Thus, we model the development triangle as a random field; the inferred correlations across Development Lags and Accident Years allow for consistent stochastic extrapolation of future claims in both dimensions. In particular, this allows a statistical investigation of the trend/correlation in both the `AY` and `DL` dimensions.

The GP framework brings multiple advantages relative to existing loss development approaches. Having a coherent statistical framework for capturing all three uncertainty layers—extrinsic observation noise, intrinsic model uncertainty and intrinsic correlation uncertainty—is the first distinguishing feature of our method. This combination offers a rigorous way to describe the predictive distribution of ultimate claims $CC_{p,Q}$. In particular, we are able to generate non-parametric, empirical distributions of any subset of $CC_{p,q}$'s. To this end, the basic GP structure implies that a forecasted ILR $Y_{p,\ell}$ is normally distributed. When modeling ILRs, the additive form in (1.5) implies that ultimate cumulative claims $CC_{p,Q}$ is also normally distributed. However, incorporating hyper-parameter uncertainty leads to a mixture-of-Gaussians distribution of $L_{p,\ell}$ and hence a much more flexible description for the tails of $CC_{p,Q}$.

Second, the spatial structure of GPs allows to generate consistent multi-period trajectories of future losses, offering a full uncertainty quantification not just of a specific $CC_{p,q}$ but of the entire vector $CC_{p,\cdot}$. Such joint draws of the full ILR path are necessary to capture the covariance structure not only of future capital requirements and risk allocation but also the timing of the emergence of claim payments. Many existing models are either unable to generate full scenarios for the time-path ($CC_{p,\cdot}$) or would generate wildly infeasible ones. For instance, since the Mack (1993) model only assumes that each LDF is log-normal and is otherwise i.i.d., consecutive draws of $CC_{p,q}$ as q changes will almost certainly lead to nonsensical non-monotone cumulative paid claims.

Third, we discuss coherent ways to adjust the GP model to respect the constraints of the development data, in particular the declining variance as losses develop, asymptotic convergence to ultimate losses, and paid incremental claims that are almost always positive. These constraints are especially important for far-into-the-future forecasts. To implement them, we built a custom *hurdle* model, augmented with virtual observations. As far as we know, this is a new methodology for GP surrogates of truncated data and is of independent interest to the GP community. Let us remark that in extrapolation tasks, which are necessarily mostly assumptions-driven, professional judgement can be conveniently encapsulated into the GP model via both the mean m and the spatial kernel structure. Moreover,

the corresponding forecast accuracy is then neatly encoded in the GP predictive variance, transparently showing to what extent is the forecast data-driven or assumption-based.

Fourth, the spatial structure extends naturally to analysis of multiple triangles, and hence brings a consistent way to handle multivariate losses. In Section 4.5 we show how GPs offer a ready-made mechanism to borrow strength (i.e. actuarial credibility) from experience of related firms.

From the empirical side, we systematically check the efficacy of point estimates and distributions across a wide range of business lines and companies based on the paid claims in the full NAIC database. In particular, we present a variety of assessment metrics that go beyond the standard root mean squared error measures to assess the accuracy of the predictive distribution and predictive intervals (CRPS, NLPD and K-S metrics, see Section 4.1). To our knowledge, these are new tools for uncertainty quantification of loss development and yield a detailed aggregate analysis.

Relative to Lally and Hartman (2018) who also investigated GP's for loss emergence, we note several key differences in the setups. First, LH work with cumulative losses, only mentioning incremental losses in passing. In our opinion, this is much less clear conceptually since it is difficult to disentangle the intrinsic dependence from cumulating claims from a true spatial correlation across `DL` and `AY`. Second, in LH the model for $CC_{p,q}$ has no provision for ensuring monotonicity or convergence to full development. Instead, LH address this nonstationarity through input warping which reduces $Var(CC_{p,q})$ as q grows. Lack of any monotonicity constraints implies that when computing predictive distribution of ultimate losses, implausible values of $CC_{p,q}$ are averaged. Third, LH eschew the issue of noisy losses—in their model there is σ^2 (taken to be constant) but it essentially serves as a numerical regularization device, bundling the randomness of claim emergence with the *average* pattern of loss development. In contrast, our approach distinguishes between the "true" loss development pattern and the observations. Fourth, LH only present a very limited study of 3 triangles; our analysis of the whole NAIC dataset offers much more comprehensive conclusions.

Finally, we mention that other Bayesian spatial models have been proposed to represent the loss development surface; see for instance Gangopadhyay and Gau (2003) who proposed to use bivariate smoothing splines combined with a Gibbs sampling. However, a spline model is less suited for extrapolation (as it does not allow specifying a prior mean function) which is the main task in completing the triangle, and is furthermore more awkward for uncertainty quantification, necessitating the use of expensive MCMC compared to the parsimony of (2.3).

Our R and `Stan` code is available for inspection via a public repository at `https://github.com/howardnewyork/gp_ilr`. That folder also contains the dataset described below and a brief guide through a README file.

1.5 Data Set

For our dataset we use the NAIC database. This database was originally assembled by Meyers and Shi (2011) who built it using Schedule P triangles from insurer NAIC Annual Statements reported in 1997. These Annual Statements contain insurer-level run-off triangles of aggregated losses by line of insurance. The data in Schedule P includes net paid losses, net incurred losses (net of reinsurance), and net premiums. Using subsequent annual statements from 1998–2006, the triangles were completed; a complete description of how it was constructed and how the insurers were selected, is available on the CAS website (Meyers and Shi).

The rectangles encompass 200 companies across 6 different business lines (not all companies have all lines): Commercial Auto `comauto` (84 companies), Medical Malpractice `medmal` (12), Private Passenger Auto `ppauto` (96), Product Liability `prodliab` (87), Other Liability `othliab` (13) and Workers Compensation `wkcomp` (57). Each rectangle consists of 10 accident years and 10 development years, covering accident years 1988-1997 and development lags 1 through $Q = 10$, covering the period 1988 to 2007 (for the 10th development year for claims originating in 1997). To construct our training set we use the upper triangle, i.e. the data as it would appear in 1997, for a total of 45 LDF observations per triangle and 55 ILR observations. From the provided data, we use only those companies that have a full set of 10-years of data without any missing fields. We also exclude companies that have negative premiums in any year. Throughout, we focus on *paid* claims.

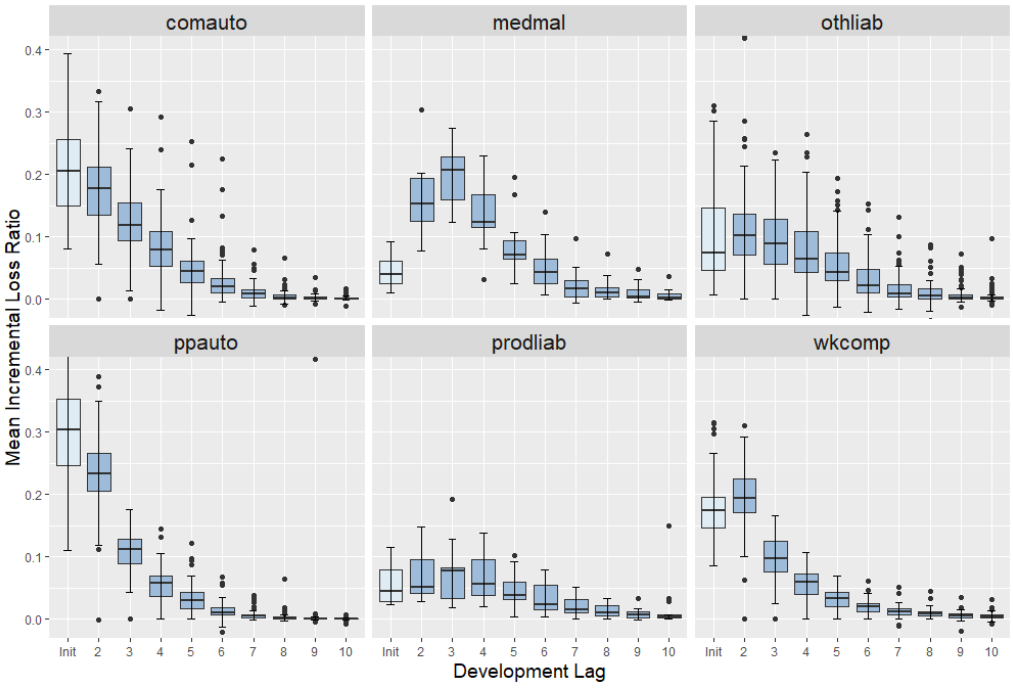

Figure 1: Distribution of ILRs for each of the six business lines. The leftmost boxplots show the distribution of the 'Init' first-year loss ratio $CC_{\cdot,1}/P_{\cdot}$ and the rest are for $L_{\cdot,q}$ for $q = 2, \ldots, 10$. To better visualize the bulk of the data, we clip the y-axis at $ILR \in [0, 0.4]$, which hides a few outliers for `comauto` and `ppauto` that have ILRs as high as 70%.

Figure 1 shows the typical shape of incremental loss ratios across different business lines (see also Table 2 in Appendix C that shows a typical triangle in terms of raw $CC_{p,q}$ and the corresponding ILRs $L_{p,q}$). The boxplots show the distribution of $L_{\cdot,q}$ across the companies of a given line. We observe that as expected ILRs tend to decrease in Lag (with the notable exception of `medmal` where $ILR_{p,\cdot}$ tends to peak at $q = 3$, and `prodliab` where ILRs are essentially flat for the first 3 years). The auto insurance triangles tend to develop by $q = 8$ or so, however all other lines exhibit a significant proportion of positive ILRs even at $q = 10$. We also observe that there is a nontrivial, frequently non-monotone, pattern to the dispersion of $L_{\cdot,q}$ which implies the need for a careful statistical modeling of $\sigma_{p,q}$.

2 Gaussian Processes for Development Triangles

2.1 Gaussian Processes Overview

Gaussian Process modeling (see e.g. Rasmussen and Williams (2006); Forrester et al. (2008)) is a nonparametric machine learning framework to probabilistically represent input-output relationships. The GP has proven to be one of the best-performing methods for statistical learning, providing flexibility, tractability and a fast-growing ecosystem that incorporates many extensions and computational enhancements. Moreover, compared to other machine learning methods, such as neural networks or random forests, GPs are highly suited for actuarial applications thanks to their intrinsic probabilistic structure that provides fully stochastic forecasts.

In the description below we adopt the abstract perspective of inputs x and outputs y. Recall that our input cells are bivariate $x = (\texttt{AY}, \texttt{DL})$ and the outputs are certain transformations of observed losses so far.

In traditional regression, including linear, generalized linear and generalized additive modeling, f in (1.4) is assumed to take on a known parametric form. Under the GP paradigm, f is deemed to be latent and modeled as a random variable. Formally, a GP $f \sim GP(m, C)$ is defined as a set of random variables $\{f(x) | \, x \in \mathbb{R}^d\}$, where any finite subset has a multivariate Gaussian distribution with mean $m(\cdot)$ and covariance $C(\cdot, \cdot)$. A key difference between a GP and a standard multivariate normal distribution is that the latter is defined by a mean vector and covariance matrix, whereas a GP is defined by a mean function and covariance function.

GP regression recasts the inference as computing, using Bayes rule, the posterior distribution of f conditional on the observed data $\mathcal{D} = (\mathbf{x}, \mathbf{y})$, where $\mathbf{x} = \{x^i | i = 1, \ldots, N\}$, $\mathbf{y} = \{y^i | i = 1, \ldots, N\}$. A key insight of the theory is that if the prior for f is a GP, then so too is the posterior f_* over new inputs x_*, namely $f_*(x_*)|\mathcal{D} \sim GP(m_*(x_*), C_*(x_*, x_*))$. In the standard setup, closed form expressions can be obtained for the posterior mean $m_*(x_*)$ and the posterior covariance $C_*(x_*, x_*)$. If ϵ^i's are normally distributed, then the posterior for the outputs, y^i_*, will also be a GP.

To construct a GP model, the user must specify its (prior) mean function $x \mapsto m(x)$ and the covariance structure $C(x, x')$, as well as the noise structure, i.e. the likelihood

$p(Y(x)|f(x))$. This is usually done by picking a parametric family for each of the above, so that inference reduces to learning the *hyperparameters* governing the structure of $\{f_{p,q}\}$. There are two broad approaches to fitting a GP to data. The first is an optimization based approach, and the second is a Bayesian approach. The optimization approach entails using maximum likelihood estimation to solve for the hyperparameters underlying the GP mean and covariance functions. The Bayesian approach requires establishing priors for each of the parameters and then solving for the posterior distribution of each of the hyperparameters. This is done by Markov Chain Monte Carlo, generating a chain of samples drawn from the marginal distribution of each of the parameters. These samples can then, in turn, be used to generate a posterior distribution for f_* on the training inputs x^i as well as new inputs x^i_*. The primary advantage of the optimization is that the method is fast compared to the Bayesian approach. The Bayesian approach is however more flexible, and with judicious choices of priors can produce substantially more robust solutions. In sum, the steps involved in modeling with a GP entail:

1. Pick a prior mean and covariance function, plus noise specification;
2. Use MLE or a Bayesian approach to solve for the hyperparameters;
3. Use multivariate normal identities to condition on data $\mathcal{D}$ + conditional simulations to generate forecasts of $CC_{p,q}$ (both in- and out-of-sample);
4. Test the goodness-of-fit of the model;
5. Try alternate mean and covariance functions, repeat steps 2-4.

Practically, as our primary computing environment we have utilized the probabilistic programming language `Stan` (Carpenter et al., 2017) to build a fully-Bayesian GP model. Stan uses Hamiltonian Monte Carlo (HMC) as its core algorithm for generating a chain of samples drawn from the marginal distribution of each of the parameters. These samples can then, in turn, be used to generate a posterior distribution for f_* on the training inputs x^i, as well as new inputs x^i_* (Betancourt, 2017). In addition, we also compared to alternative implementation of GPs in R via the `DiceKriging` package (Roustant et al., 2012) that employs MLE-based hyperparameter optimization.

2.2 Determining the Posterior of a Gaussian Process

As mentioned previously, if the prior $f \sim GP(m(\cdot), C(\cdot,\cdot))$, then the posterior, f_* over new inputs x_*, is a multivariate normal random vector and has a closed form expression. That is, for any cell x_*, $f_*(x_*)$ is a random variable whose posterior given $\mathcal{D} \equiv \{\mathbf{x}, \mathbf{y}\}$ is:

$$f_*(x_*)|\mathbf{y} \sim \mathcal{N}\Big(m_*(x_*), s^2(x_*)\Big) \tag{2.1}$$

$$m_*(x_*) = m(x_*) + C_*\mathbf{C}^{-1}(\mathbf{y} - m(\mathbf{x})) \tag{2.2}$$

$$s^2(x_*) = C(x_*, x_*) - \mathbf{C}_*(x_*)\mathbf{C}^{-1}\mathbf{C}_*(x_*)^T, \tag{2.3}$$

where the $N \times N$ matrix covariance matrix $\mathbf{C}$ and the $N \times 1$ vector $\mathbf{C}_*(x_*)$ are

$$\mathbf{C} := \begin{bmatrix} C'(x^1, x^1) & C(x^1, x^2) & \dots & C(x^1, x^N) \\ C(x^2, x^1) & C'(x^2, x^2) & \dots & C(x^2, x^N) \\ \vdots & \vdots & \ddots & \vdots \\ C(x^N, x^1) & C(x^N, x^2) & \dots & C'(x^N, x^N) \end{bmatrix}, \ \mathbf{C}_*(x_*)^T := \begin{bmatrix} C(x_*, x^1) \\ C(x_*, x^2) \\ \vdots \\ C(x_*, x^N) \end{bmatrix}, \tag{2.4}$$

and $C'(x, x) = C(x, x) + \sigma^2(x)$. Consequently, the point forecast at x_* is $m_*(x_*)$ and $s^2(x_*)$ provides a measure of uncertainty (akin to standard error) of this prediction. The role of s^2 can be thought of as (spatial) credibility—posterior variance is low when x_* is in the middle of the training set $\mathbf{x}$ and grows as we extrapolate further and further away from $\mathbf{x}$. Our interpretation is of $s^2(x_*)$ as the intrinsic model uncertainty, which is then overlaid with the observation noise $\sigma^2(x_*)$, so that the predictive distribution of $Y(x_*)$ is

$$Y(x_*) \sim \mathcal{N}\left(m_*(x_*), s^2(x_*) + \sigma^2(x_*)\right) \tag{2.5}$$

where the additive structure is thanks to the independence between the intrinsic uncertainty of the ILR surface and the extrinsic noise in realized paid losses. Note that (2.1) can be vectorized over a *collection* of inputs $\boldsymbol{x}_*$, providing a multivariate Gaussian distribution for the respective $Y(\boldsymbol{x}_*)$. In contrast to Chain Ladder-like methods where loss development is done step-by-step, with a GP model we can complete the entire loss square (or any subset thereof) in a single step through the same universal multivariate Gaussian sampling. From another perspective, the GP posterior variance (2.3), which is specified indirectly through the spatial smoothing imposed by $C(\cdot, \cdot)$, is to be contrasted to the direct specification of the correlation between $CC_{p,q}$ and $CC_{p',q'}$ in the multivariate Gaussian models (Wüthrich and Merz, 2008, Section 4.3.2).

2.3 A conceptual Introduction to Gaussian Processes

The covariance structure captures the idea of spatial dependence: for any cells i and j, if x^i and x^j are deemed to be "close" or in the same "neighborhood", then we would expect the outputs, y^i and y^j, to be "close" too. This idea is mathematically encapsulated in $C(\cdot, \cdot)$: the closer x^i is to x^j, the larger the covariance $C(x^i, x^j)$ and hence knowledge of y^i will greatly affect our expectations of y^j. For example, model factors for consecutive Accident Years are expected to have high covariance, whereas factors 10 years apart would likely have lower covariance. Learning the covariance function (through e.g. likelihood maximization or a Bayesian procedure) quantifies this relationship, ultimately allowing for highly non-linear modeling of claims by location.

Among many options in the literature for the covariance function, arguably the most popular is the squared exponential kernel,

$$C(x^i, x^j) = \eta^2 \exp\Big(-\frac{1}{2}(x^i - x^j)^T \boldsymbol{\rho}^{-2} (x^i - x^j) \Big), \tag{2.6}$$

where $\boldsymbol{\rho}$ is a diagonal matrix. Traditionally, diagonal elements of the matrix $\boldsymbol{\rho}$ are known as the *lengthscale* parameters, and η^2 as the signal variance. Together they are referred to

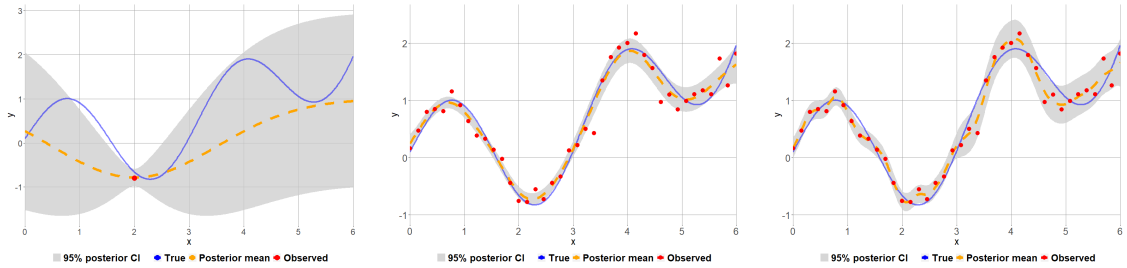

Figure 2: Training a GP model for a toy 1-D example. Dashed orange line is posterior GP mean $m_*(x)$, grey band is the 95% credible interval $\pm 1.96 s(x)$. *Left panel:* a single training input (x^1, y^1) (red dot). The GP reverts back to its prior far away from x^1; *Middle:* 40 training inputs (red dots) and $\rho = 1.5$. *Right:* same 40 training inputs with shorter lengthscale $\rho = 0.5$.

as hyper-parameters. While the lengthscale parameters determine the smoothness of the surface in the respective dimension, η^2 determines the amplitude of the fluctuations.

To provide intuitive insight into how a simple GP can model a relatively complex non-linear function, we show an example where the underlying data is generated from the following distribution (note input-dependent noise)

$$Y(x) \sim \mathcal{N}\left(0.1(x-1)^2 + \sin(2x), 0.1 \cdot 1_{\{x<3\}} + 0.25 \cdot 1_{\{x \geq 3\}}\right).$$

To start, we could arbitrarily assume for the GP prior a mean function $m(x) = 1$ and a covariance function as in (2.6) with its i, j^{th} entry defined via $C_{i,j} = \eta^2 e^{-(x^i - x^j)^2/\rho^2}$, $\eta = 1, \rho^2 = 1.5$. Now suppose our training data consists of a single point $\mathcal{D} = \{x^1 = 2, y^1 = -0.8\}$ with $\sigma^\epsilon(x^1) = 0.1$ being the observation noise. The observation at $x^1 = 2$ provides "strong" information for future measurements in that vicinity, and weak information for measurements far away. The covariance values of C for new inputs will be close to 1 near $x = 2$ but close to zero for x values far away from 2, e.g. $C(1.5, x^1) = 0.846$, but at $x = 5$, $C(5, x^1) = 0.0024$. Thus we are more confident in our predictions for inputs x_* close to 2, then further away, reflected in the "sausage-like" posterior 95% credible interval shown in the left-most graph in Figure 2. Note that on the right side of the plot, the lack of any nearby data points implies that approximately $f_*(x_*) \sim \mathcal{N}(1, 1) \equiv \mathcal{N}(m, \eta^2)$.

When there are more data in our training set, the model gets "stronger" in the vicinity of the data points, see the middle panel in Figure 2 where we train on 40 data points. Here the trained GP model (dashed orange) matches quite well the true response (solid blue). We see that even with a very simple kernel definition (2.6), we are able to model a complex non-linear model with noisy measurement. As more data is received on which to train the model, the credible intervals shrink.

To give a sense of the effect of the lengthscale hyperparameter on the model, we fit another GP to the same data maintaining the square exponential kernel. The first model has $\rho^2 = 1.5$, and the second has shorter lengthscale $\rho^2 = 0.5$. Smaller ρ decorrelates the observations faster and hence makes f_* more "wiggly", see the right panel of Figure 2. Here the smaller ρ clearly overfits visually, confirming the need to have a rigorous procedure for

picking the hyperparameters. Below we discuss how to specify the parametric structure or kernel of the covariance function, how to learn the parameters of the kernel, how to use GPs to conduct extrapolation, not just interpolation, tasks and what special features are required for modeling paid loss development data.

2.4 Mean Function

The prior mean function $m(\cdot)$ stands in for the prior belief about the structure of f in absence of observed data. This resembles Clark (2003) approach that prescribes a parametric growth curve for $CC_{p,\cdot}$ (or for $I_{p,\cdot}$). The role of the mean is to guide the inference by detrending observations against a known structure. One can also combine such parametric trend assumptions with the GP framework through the so-called Universal Kriging that takes $m(x) = \sum_i \beta^i \phi_i(x)$, where ϕ_i's are the prescribed basis functions, and simultaneously infers the trend coefficients β^i in conjunction with solving (2.1).

As can be seen from the example shown in Figure 2, a posterior GP will revert to the prior GP or will be more heavily influenced by the prior GP when the projection is conducted for inputs that differ materially from the observed data. An alternative for guiding the data, is to dispense with the mean function (by using $m(x) = 0$) and adjust the kernel to incorporate a linear regression component. For example, in a Bayesian linear regression setting, $x \mapsto Y(x)$ is a linear function, $Y(x) = ax+b$, and the priors for unknown a and b are $a \sim \mathcal{N}(0, \sigma_a^2), b \sim \mathcal{N}(0, \sigma_b^2)$. Then $\mathbb{E}[Y(x)]$ can be trivially seen to be zero and $\mathrm{Cov}(Y(x^1), Y(x^2))$ for any inputs x^1, x^2 is determined as:

$$\begin{aligned} Cov\left(Y(x^1), Y(x^2)\right) &= \mathbb{E}[Y(x^1)Y(x^2)] - \mathbb{E}[Y(x^1)] \cdot \mathbb{E}[Y(x^2)] \\ &= \mathbb{E}\left[(ax^1 + b) \cdot (ax^2 + b)\right] = \sigma_a^2 \cdot x^1 x^2 + \sigma_b^2. \end{aligned}$$

Thus, a Bayesian linear regression can be modeled as a GP (although rather inefficiently), with mean function $m(x) = 0$ and covariance $C^{(lin)}(x, x') = \theta \cdot xx' + \sigma^2$, known as the linear kernel.

It can be shown (Rasmussen and Williams, 2006, Ch. 4) that a quadratic function can be modeled in a GP setting by multiplying together two linear kernels. Furthermore, a range of more complicated functions can be modeled as combinations (namely adding or multiplying) of simple kernels, such as linear or periodic kernels. In such a way, compound kernels provide structure to GP regression, which can be particularly useful for conducting extrapolation rather than just interpolation predictions. Details of working with compound kernels can be found in (Duvenaud et al., 2013; Rasmussen and Williams, 2006). By creating compound kernels, a GP can model non-linear data locally and then blend into a linear or quadratic regression model for distant extrapolations.

In this paper, we employ the compound kernel:

$$C(x, x') = C^{(SqExp)}(x, x') + C^{(lin)}(x_{AY}, x'_{AY}) + C^{(lin)}(\log(x_{DL}), \log(x'_{DL})). \tag{2.7}$$

The squared-exponential component will dominate in areas where there are lots of training data. For distant extrapolations, the former will trend towards zero and so the linear

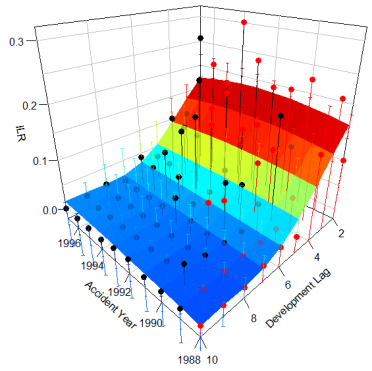

Figure 3: 3-D view of the loss square for a representative `comauto` dataset with red dots indicating the training $L_{p,q}$'s and black dots the bottom triangle to be completed. The smooth surface shows the forecasted completion $m_*(x)$ using a plain GP ILR model with constant observation noise. The error bars denote $\pm 1.96 s(x)$ from (2.3) which is the intrinsic Gaussian-based posterior model uncertainty of $f_{p,q}$.

kernel will be dominant. Other kernel families, such as Matern, could also be straightforwardly used. In our experiments, the choice of the kernel family plays a secondary role, in part because the triangle is in tabular form and therefore does not possess the fine local correlation structure that is most affected by kernel choice.

3 Structural Constraints of ILRs

The left panel of Figure 3 shows a basic GP fit of ILR for a typical triangle. It illustrates the inherent smoothing of the GP approach which converts noisy observations from the upper triangle, $L_{p,q}$ (red dots), into a smooth latent surface of the modeling factor, here ILR. It also highlights several key features of ILRs:

- As $q \to Q$, $f_{p,q} \to 0$ generally decreasing monotonically;
- Cumulative paid claims are generally increasing so that the true ILR should satisfy $f_{p,q} \geq 0$ for any cell;
- As $q \to Q$, $L_{p,q} \to 0$, generally monotonically, so there is little intrinsic and extrinsic uncertainty for long lags;
- As $q \to Q$, $Var(L_{p,q})$ decreases rapidly, so ILRs are highly volatile for short development lags, but have minimal variance for longer lags;

These features create the double challenge of constraints. First, the generated predictive distributions of $L_{p,q}$ must satisfy these constraints. In particular, it implies that the observation noise ϵ must be structured appropriately. Second, it might be desirable to

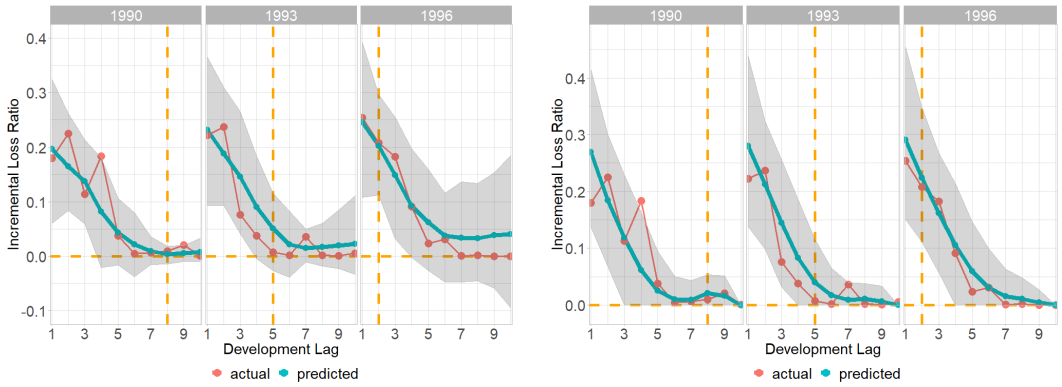

Figure 4: Predictive distribution of the incremental loss ratios $L_{p,q}$ for three representative accident years. The vertical dashed line indicates the edge of the training triangle. We show the predictive mean and 95% credible interval to be validated against the actual observations. Left: basic GP ILR model. Right: Hurdle + virtual points GP model for a representative `comauto` triangle. The predictive distribution collapses to point mass at $L_{p,Q} = 0$ at lag $q = 10$. The hurdle likelihood ensures that $L_{p,q} \geq 0$ for all cells.

have the latent $\{f_{p,q}\}$ also satisfy the constraints. For example, the assumption of having a Gaussian prior on $f_{p,q}$ might be challenged.

3.1 Overall Procedure

For the latent surface $f(x)$, after trying multiple approaches, we recommend using a plain GP structure. In particular, we do not impose any constraints on the latent surface, for example allowing $m_*(x) < 0$ for ILR modeling. At the same time, it is critical to enforce that the predictive mean/standard deviation $m_*(x) \to 0, s_*(x) \to 0$ as the Lag increases. Due to the spatial structure of the GP, without this constraint, forecasting ILR far into the future, i.e. far bottom-right of the completed triangle will intrinsically reduce to the prior, i.e. $f_*(x_*) \sim \mathcal{N}(m(x_*), \eta^2)$. In turn, this leads to extremely wide predictive intervals, as the model would suggest a high degree uncertainty about the posterior distribution of $f_*(x_*)$ far into the future. Rather than more complex work-arounds such as non-constant GP process variance $\eta^2(x)$, we found that the combination of a hurdle model, decreasing σ_q and virtual observations, all described in Section 3.2 below, ensure that $L_*(x_*) \simeq 0$ at large lags, see right panel of Figure 3. This also offers a convenient (albeit ad hoc) way of handling the desired mixed distribution of realized ILRs for large q.

3.1.1 What to Model

Because there are many quantities that might be derived from the developed losses, the question of which quantity to model is non-trivial. In our analysis, we found that incremental loss ratios work best. This is partly justified by the fact that the spatial dependence assumption (in particular across accident years) is most appropriate.

In contrast, other approaches have worked with unscaled quantities, such as the cumu-

lative losses $CC_{p,q}$ or the incremental claims $I_{p,q}$. For example, the Meyers (2015) LCL approach postulates that $CC_{p,q}$ is log-normal with mean $\alpha_p+\beta_q$ and standard deviation σ_q. Lally and Hartman (2018) also directly model $CC_{p,q}$. The Overdispersed Poisson (ODP) method is based on modeling $I_{p,q} \sim OPoisson(i(p,q))$.

The choice of the object to model strongly impacts the shape of the respective latent surface and the appropriate noise distribution. Incremental paid loss amounts $I_{p,q}$ tend to be skewed to the right and are occasionally negative, while $CC_{p,q}$ is much more volatile since it is strongly affected by the overall claim volume. The fact that claims settle and hence eventually stabilize leads to different asymptotics: $F_{p,q} \to 1$, $L_{p,q} \to 0$, $I_{p,q} \to 0$ and $CC_{p,q} \to CC_{p,Q}$.

3.1.2 Variable Transformations

A natural way (analogous to a link function in generalized linear models) to enforce $L_{p,q} \geq 0$ is via a log-transform. However, we found that the reverse exponentiating to convert predictions to original scale tend to be highly unstable with GPs, in particular generating very high predictive variance. A similar cause is our preference towards modeling ILRs (that lead to additive structure when computing ultimate claims) relative to LDFs that come with a multiplicative structure. Indeed, our experiments suggest that properly building a GP model for LDFs is quite challenging, as the models tend to generate excessive uncertainty in LDF at large lags. For example, even a hurdle LDF model will occasionally lead to nonsensical predictive mean of $CC_{p,q}$ that is multiple orders of magnitude larger than what is reasonable.

3.2 Noise Modeling

To ensure that $Var(L_{p,q})$ progressively decays as $q \to Q$, we take the variance of $\epsilon_{p,q}$ to depend on q. The latter feature forces the estimated ILR surface $f_{p,q}$ to closely match observed $L_{p,q}$'s for long development lags to capture the full development of losses at Q. At the same time, we allow $L_{p,q}$ to deviate far from observations for short lags. To achieve this, the observation variances $\sigma^2_{p,q}$ are viewed as a non-parametric function of the `DL` q. The respective hyper-parameters are learned with other GP hyper-parameters, assuming a decreasing structure in `DL`. Our experiments suggest that an informative prior for σ_q is crucial for the MCMC chains to function well. The right panel of Figure 6 shows the respective posterior distribution where we see that the fitted σ_q's rapidly approach 0 as q increases. This makes the forecasts get tighter around $m_*(x)$ as Lag increases, reflecting reduced uncertainty in incremental losses.

Remark 1. For cumulative models, $Var(CC_{p,q})$ increases in q and should approach some ultimate asymptotic value. For incremental models, the same effect is achieved implicitly, since the uncertainty in ultimate losses is primarily based on the product/sum of individual Y-terms that progressively become less and less stochastic.

Furthermore, to reflect that generally $L_{p,q} \geq 0 =: \underline{y}$, we had success with the following

Gaussian hurdle type observation model:

$$L(x) \sim \mathcal{N}\left(m_*(x), \sigma^2(x)\right) \vee \underline{y}, \tag{3.1}$$

where $a \vee b = \max(a, b)$. Thus, future ILRs are first sampled from the Gaussian predictive distribution and then are rounded up to zero to directly enforce the non-negativity constraint. Note that it makes the predictive distribution a mixture of a truncated Normal and a point mass at $\underline{y} = 0$; the resulting $L_{p,q}$ is skewed and non-centered, i.e. $\mathbb{E}[L(x)] > m_*(x)$. The respective likelihood $L(x)|f(x)$ during the inference step is a mixture of a Gaussian likelihood if $L(x) > \underline{y}$ and a Bernoulli likelihood $\mathbb{P}(L(x) = \underline{y}|f(x)) = \Phi(\underline{y}; f(x), \sigma^2(x)) =: h(x)$, that we term the hurdle probability.

The hurdle probability takes care only of the left tail of $L_{p,q}$. It is also important to ensure that $s(x_*) \simeq 0$ for x_* at the bottom-right of the completed square, otherwise large $s(x_*)$ implies large posterior uncertainty in $f_*(x_*)$ and hence very wide (right-skewed) predictive intervals even with (3.1) in place. To do so, we add *virtual* observations at $q = 11$, namely that losses fully develop by $Q = 10$. Computationally we simply augment our training data set with $L_{p,Q+1} = 0 \forall p$. Virtual data points allow for incorporation of professional judgement by including information that lies outside the available loss triangle training data set.

Remark 2. We also tried $\sigma_q = StDev(L_{\cdot,q})$ based on the empirical standard deviation of the $Q-q$ available observations at Lag q, which is related to the stochastic kriging approach (Ankenman et al., 2010). However, this method is highly unstable except for the first couple of `DL`'s as otherwise there are too few observations to estimate the standard deviation with any certainty. Yet another possibility is to adjust $\sigma^2_{p,q}$ based on the credibility of individual loss cells.

3.3 Models Tested

For our tests we have run the following three GP-based ILR models:

- ILR-Plain: Stan Bayesian GP model with a compound linear kernel (parametrized by trend coefficients θ) that fits $m(x) = \theta_1 \texttt{AY} + \theta_2 \log \texttt{DL}$. We also fit Lag-dependent observation variances σ^2_q. We use (2.7) and take the following weak hyperpriors:

$$\rho \sim InvGamma(p_{\rho,1}, p_{\rho,2}), \qquad \eta^2 \sim \mathcal{N}(0, p_{\eta^2}), \qquad \sigma \sim \mathcal{N}(0, p_\sigma), \qquad \theta \sim \mathcal{N}(0, p_\theta). \tag{3.2}$$

  To set the hyperpriors for ρ, we follow the exposition by Betancourt (2017). The GP model becomes non-identified when the lengthscale parameter is either below the minimum lengthscale of the covariate or above the maximum lengthscale of the covariate. Our training inputs are equally spaced and the min and max lengthscales are $ls_{min} = 1$ and $ls_{max} = 10$. We then use these bounds to place light tails for the transformed lengthscale ρ by tuning $p_{\rho,1}, p_{\rho,2}$ to obtain $\mathbb{P}(\rho < ls_{min}) = \mathbb{P}(\rho > ls_{max}) = 0.001$, yielding a more informative prior. We also scale the training inputs to have zero mean and standard deviation one.

- ILR-Hurdle: Bayesian GP for ILRs with a hurdle observation likelihood (3.1) and constrained mean function;
- ILR-HurdleVirt: Bayesian GP for ILRs with a hurdle likelihood and virtual observations at $q = 11$;

For all models we utilize 8 MCMC chains, with a burn-in of several hundred transitions and generating at least 1000 posterior samples.

To demonstrate performance of GP models relative to "vanilla" loss development frameworks we also employed the `ChainLadder` (Gesmann et al., 2018) R package to construct a Mack stochastic Chain Ladder ("Mack CL") which fits log-normal distributions to LDF $F_{p,q}$. Mack CL is equivalent to an overdispersed-Poisson generalized linear model for incremental claims $I_{p,q}$.

Remark 3. In Appendix A we also report the results from using the Bootstrap Chain Ladder in `ChainLadder` (Gesmann et al., 2018) which replaces the parametric LDF modeling of Mack CL with a nonparametric bootstrap.

4 Empirical Results

4.1 Performance Metrics

In general, all loss reserve models are trained on the upper left portion of the loss square, with the model being used to project the bottom right portion. Since loss development forecasts are used for multiple purposes, we identify an appropriate performance metric for each likely broad application of the model. (Below we use R to denote a generic random quantity being validated against its realizations r.) This allows for the comparison among GP implementations as well as against more standard models. The intended application of loss reserve models with the matching performance metrics are now described. A key feature that we emphasize is the ability of GPs to generate full probabilistic forecasts, i.e. the entire predictive distribution, which allows for more sophisticated tests relative to methods that only return point forecasts.

1. Best Estimate Reserves: A primary task of loss reserve models is to establish a best estimate reserve for unpaid claims. To measure the accuracy of a model we test the ultimate projected cumulative claims by calculating the root mean squared error ("RMSE") of the ultimate projected cumulative losses, namely $R = \Sigma_p CC_{p,Q}$ against the actual amounts. We also calculate the "weighted" RMSE on cumulative paid loss ratio amounts, namely $R = (\sum_p CC_{p,Q})/((Q-1) \cdot \sum_p P_p)$, where P_p are the premiums for AY p. The latter metric down-weights the larger companies that have higher P_p's and provides a more interpretable result. The RMSE captures the accuracy of the point forecast in terms of minimizing bias and controlling for variance and are aggregated (by averaging) across triangles, grouped by line-of-business.

2. Cash Flow Projections: A key financial task is to conduct cash flow projections for the emergence of claims payments. This is a core component of an asset-liability management program. To do this, a preliminary step is to calculate the "n-step" ahead cumulative loss ratios, namely $R^{(n)} = \sum_p CC_{p,Q-p+n+1}/P_p$. We then calculate the RMSE on this value. We expect $R^{(n)}$'s to increase in n as uncertainty necessarily rises for longer forecast horizons. Thus, the n-year-ahead RMSE offers a more granular assessment and corrects for the tendency of the overall RMSE to be unduly influenced by the errors in the bottom-right of the triangle.

3. Risk Capital: Risk capital is required to cover reasonable worst case loss scenarios, and the accuracy of the left hand tail of the ultimate loss distribution is thus important for this measure. A good model, for example, should show 5% of the empirical cumulative claims, $CC_{p,Q}$, being less than the 5% predictive quantiles. To assess the tails of the predictive distribution we report the coverage probability: the empirical frequency that the realized losses land within the α%-band of the predictive distribution. Below we test for 90% nominal coverage, looking at the 5%-95% predictive quantiles.

4. Sensitivity Analysis: This involves analysis of reserves based on reasonable downside and upside cases. In this case, we would want the overall predictive distribution to represent a good model for the data so as to detect potential model misspecifications. This leads us to more formal statistical tests described below.

To assess the quality of the predictive distribution, we utilize proper scoring functions (Gneiting and Raftery, 2007) as implemented in `scoringRules` (Jordan et al., 2019) R package, which return quantitative scores, such that better models should have (asymptotically) lower test scores than worse models. The proper scoring function ensures that a model is penalized for forecasting too much uncertainty; simpler metrics are typically only sensitive to under-estimating the variance of CC. Two specific criteria are the Continuous Ranked Probability Score (CRPS) and the Negative Log Probability Density (NLPD). CRPS is defined as the squared difference between the predictive CDF and the indicator of the actual observation,

$$\text{CRPS}(r) := \int_{\mathbb{R}} (\hat{F}(z) - 1_{\{r \leq z\}})^2 dz. \tag{4.1}$$

For repeated measurements, CRPS can be interpreted as the quadratic difference between the forecasted and empirical CDF of R. In our case the distribution of R changes company-to-company, but we can still average CRPS's across triangles.

NLPD is a popular tool for testing general predictive distributions. For a single triangle, the NLPD is defined as

$$\begin{aligned} \text{NLPD}(r) &= -\sum_{i=T+1}^{N} \log p(y_i | y_{1:T}) \\ &= \left(\frac{r - \hat{r}}{\sigma_r}\right)^2 + \log \sigma_r^2, \end{aligned} \tag{4.2}$$

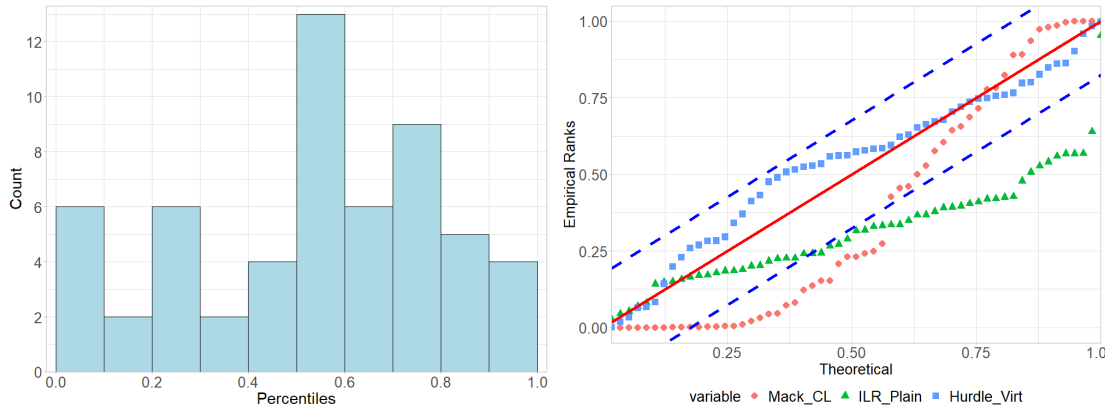

Figure 5: Left: Percentile rank of realized ultimate losses in terms of the predictive distribution of the ILR Hurdle+Virt model across 57 `wkcomp` triangles. Right: Kolmogorov-Smirnov test across three models for `wkcomp`. The dashed lines indicate the K-S test thresholds that are z_{KS} from the 45-degree line; only the Hurdle+Virt model passes the K-S test (stays between the dashed lines).

in the case where the predictive distribution is Gaussian $R \sim \mathcal{N}(\hat{r}, \hat{\sigma}_r^2)$ and where T is the number of training samples, and N is the total number of data samples. The NLPD is analogous to the likelihood function, except that lower values imply better fits of the model to the data. This score can be approximated directly in Stan using MCMC techniques, see Bürkner et al. (2020). We calculate the NLPD on the ultimate losses of each triangle, and then add up across companies.

To further validate the entire forecasted distribution of R we consider the realized quantile (i.e. "rank") of r relative to the predicted CDF. If the forecast is correct, those ranks should form a uniform distribution, since $F_R(R) \sim Unif(0,1)$, see Figure 5. More formally we implement a Kolmogorov-Smirnov test on the realized sorted percentiles $\{p_i\}$ across N triangles of a given business line. Let $\{f_i\} = 100 \cdot \{1/N, 2/N, \ldots, N/N\}$ be the respective N uniform percentiles. The K-S test rejects the hypothesis that $\{p_i\}$ is uniform at the 5% level if $D := \max_{i \leq N} |p_i - f_i|$ is greater than its critical value z_{KS} based on Kolmogorov's distribution (Carvalho, 2015). This can be visualized through a $p-p$ plot by plotting p_i's (y-axis) against f_i's (x-axis) and adding diagonals above and below the 45-degree line, see right panel in Figure 5. The same test was implemented by Meyers (2015) for $\sum_p CC_{p,Q}$ over the entire NAIC dataset.

To recap, we assess our models through multiple test metrics sorted in terms of their specificity: RMSE is only based on the predictive mean, NLPD is based on the predictive mean and variance, coverage is based on the predictive quantile, while Kolmogorov-Smirnov and CRPS is based on the full predictive distribution.

4.2 Predictive Accuracy

Table 1 lists the aggregate results for the `wkcomp` triangles (see Appendix A for the other lines). We recall that "lower is better" for all metrics besides Coverage which should be at

0.90. First, we observe a broad agreement between the two RMSE metrics (total RMSE based on ultimate cumulative losses and RMSE on the loss ratios weighted by the respective premia). At the same time, the more granular metrics signal that low RMSE is not necessarily correlated with good predictive uncertainty quantification. Indeed, while Chain Ladder achieves low RMSE scores, it fails to validate across the predictive distribution tests. Specifically, they have low coverage (so frequently actual losses are "extreme" in terms of predictive quantiles), high NLPD and very high K-S scores, failing the K-S test. This implies that their predictive distribution is not representative of the realized R_{ult}, in particular the predictive variance is "off". In contrast, GP ILR models perform much better on all of the above and hence more properly capture the predictive distribution. Indeed, in Figure 5 we see that for example the ILR Hurdle+Virt model is able to reproduce the expected uniform ranks, while the Mack CL model *underestimates* variance of $CC_{p,Q}$ so that many realizations are completely outside the predicted interval, yielding extreme ranks close to zero or one, see the right panel. This is corroborated by low coverage of Mack CL and high coverage (close to the nominal 0.90) by the GP ILR models. (In fact, ILR-Plain has too much uncertainty as its coverage is above 0.90). The last column of Table 1 shows that Mack CL does not pass the K-S test, while the ILR Hurdle models do and have K-S statistics that are more than 50% lower. The NLPD metric matches the above discussion although it is harder to interpret; we also note that CRPS suggest that ILR models are worse, likely due to being unduly influenced by outliers. Namely, if a method does not work well in a few triangles then its respective $CRPS(r)$ score would be very high, skewing the average CRPS across the entire business line.

Similar patterns are observed for other business lines. We note that GP methods have trouble dealing with the `comauto` triangles where they generate worse RMSE and other statistics, including not pssing the K-S test. One conclusion is that no one particular metric provides a full picture, so a comprehensive validation analysis is truly warranted. In Appendix B we also plot the n-step ahead RMSE across different methods. As expected, predictive accuracy drops significantly for many steps ahead, i.e. trying to forecast cumulative loss ratios 5+ years into the future leads to large RMSE.

Algorithm	Total RMSE	LR RMSE	Coverage	CRPS	NLPD	K-S
Mack CL	24726	0.049	0.509	388621	1305.6	0.306
Bootstrap CL	24895	0.052	0.544	389144	1232.5	0.292
ILR-Plain	84728	0.137	0.983	1150989	1113.5	0.244
ILR Hurdle	38638	0.081	0.912	640873	1072.1	*0.126*
Hurdle+Virt	39577	0.079	0.895	604892	1072.4	*0.142*

Table 1: Results for `wkcomp`. Algorithms that pass the Kolmogorov-Test (K-S statistic below the threshold of $z_{KS} = 0.1767$) are in italics.

4.3 Interpreting Gaussian Process Models

Relative to classical models, the parameters in a GP framework are rather different. The key parameters are the length-scales ρ and the Lag-dependent observation variance σ_q. Figure 6 displays the posterior of these two across three different business lines in our dataset. Specifically, we show the distribution of the MCMC posterior means for the lengthscales ρ. We observe that in terms of AY, `wkcomp` has the largest ρ_{AY}, i.e. the most time-stationary ILRs, while `medmal` has much lower ρ_{AY} implying that different Accident Years can yield quite distinct ILRs for the same lag DL. A similar pattern occurs for ρ_{DL}. We observe that $\rho_{DL} \ll \rho_{AY}$ suggesting that there is less dependence (as expected) in Development Lag compared to Accident Years. We also note lower ρ_{DL} for `medmal` implying that there is limited serial correlation in ILRs even in consecutive years. The middle panel of Figure 6 shows the fitted σ_q's. As discussed, observation variance generally decreases rapidly in DL. The shape of σ_q reflects the spread in raw ILRs as captured in Figure 1. For example, the faster development of `wkcomp` claims is reflected in the respective observation variances σ_q converging to zero faster.

Finally, the right panel of Figure 6 displays the hurdle probabilities $h_q := \mathbb{P}(L_{p,q} = 0)$. The interpretation of h_q is as the likelihood that the observed ILR will be exactly zero, i.e. no new losses will take place. Generally, h_q should be increasing in q and approach 1 as losses fully develop and $CC_{p,q}$ becomes constant. We see that full development generally occurs within 7 years for `wkcomp` and `comauto` (with workers compensation converging to zero a bit faster), while `medmal` develops much slower and even at $q = 9$ there is about 5% chance that $L_{p,q} > 0$. The humped shape of ILRs for `medmal` in Figure 1 clearly manifests itself here, resulting in an inverted-hump shape of h_q's. The hurdle probabilities provide a non-parametric, model-based summary of the probability for losses to fully develop after q lags.

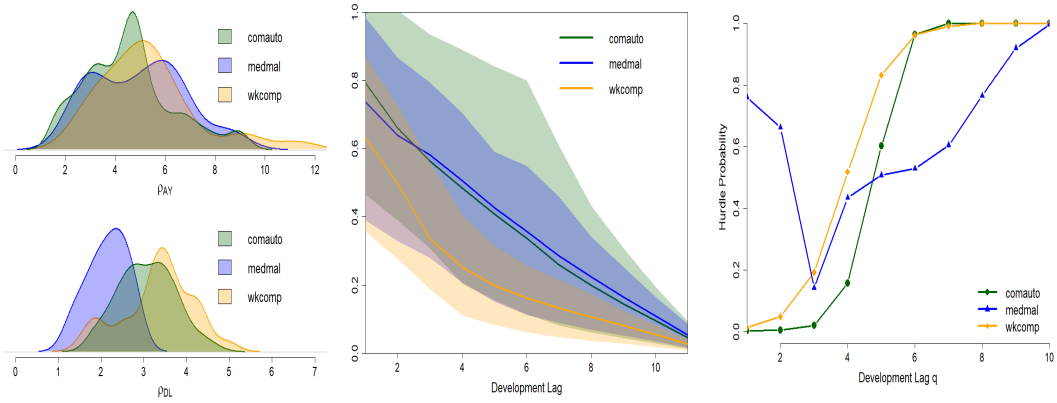

Figure 6: Lengthscales ρ_{AY} and ρ_{DL} (left panel), observation variance σ_q (middle panel) and hurdle probability h_q (right panel) (MCMC means across companies within the business line) for three representative business lines.

A major reason for adopting a fully Bayesian approach to loss development is to obtain comprehensive uncertainty quantification. The top row of Figure 7 visualizes predictive

distributions of $CC_{p,q}$ for a fixed accident year $\texttt{AY} = 1995$. In that case, there are 3 actual observed losses at $q = 1, 2, 3$ and we then display 1000 sample predictions of $CC_{1995,q}$ for $q = 4, \ldots, 10$. Recall that the GP framework directly samples the entire trajectory of the respective loss factors for $q \geq 4$ which are then converted into cumulative losses and finally into the respective empirical predictive distributions. With our Bayesian GP approach the analysis is done empirically with the MCMC samples returned by Stan.

Having direct access to the stochastic scenarios of the completed triangle in turn allows full probabilistic analysis of any given aspect of loss development, from dynamic capital allocation to risk management. This is illustrated in the bottom-left of Figure 7 where we plot the forecasted distribution of ultimate losses $R_{ult} := \sum_p CC_{p,Q}$ vis-a-vis the actual realized ultimate loss; the respective quantile is used for the rank testing and CRPS/NLPD computations. Besides the forecasts from ILR-Plain and ILR-Hurdle+Virt that match the same 1000 scenarios shown in the top row of the Figure, we also include the classical Mack CL forecast. The plot clearly conveys that the Mack CL predictive distribution is very narrow, i.e. has much lower predictive variance compared to the Bayesian GP fits, which matches its poor predictive coverage and percentile-rank score. On the contrary, ILR-Plain model over-estimates predictive uncertainty, partly due to not accounting for the intrinsic constraints on $Y_{p,q}$. Note that even if the respective distribution of R_{ult} might look reasonable, the more granular analysis of the generated trajectories of $CC_{p,q}$ in the top panel reveals the fundamental mismatch between the ILR-Plain model and the nature of loss development. Indeed, a substantial fraction of the trajectories in the top-left of Figure 7 feature either decreasing cumulative losses or sudden upward jumps in $CC_{p,q}$, neither of which are reasonable. In contrast, the trajectories in the top-right panel corresponding to Hurdle+Virt model look much more realistic. Another important observation from Figure 7 is how the correlation uncertainty leads to a non-Gaussian predictive distribution of R_{ult}, in particular we observe a strong right-skew. In contrast, the Mack CL prediction is essentially symmetric.

Our analysis demonstrates that GPs offer a flexible approach to capture loss development dependencies. They are competitive with existing methods in terms of accuracy (RMSE) and provide better predictive distributions in terms of more sophisticated variance, interval and rank tests.

4.4 Partial Bayesian Models

It can be difficult to fully dis-entangle the two intrinsic uncertainty sources of a GP model. Recall that the posterior GP variance $s_*(x)$ reflects the concept that many "true" surfaces f can be fitted based on the limited amount of observed loss data. In complement, the hyper-parameter or correlation uncertainty reflects the impossibility of fully inferring the correlation between different loss triangle cells.

To better connect to the classical Chain Ladder, we investigated partial Bayesian models that provide a recipe for learning the true modeling factors and then super-impose stochastic forecasts. In this setup, we view the GP MVN conditioning formula (2.1) as a

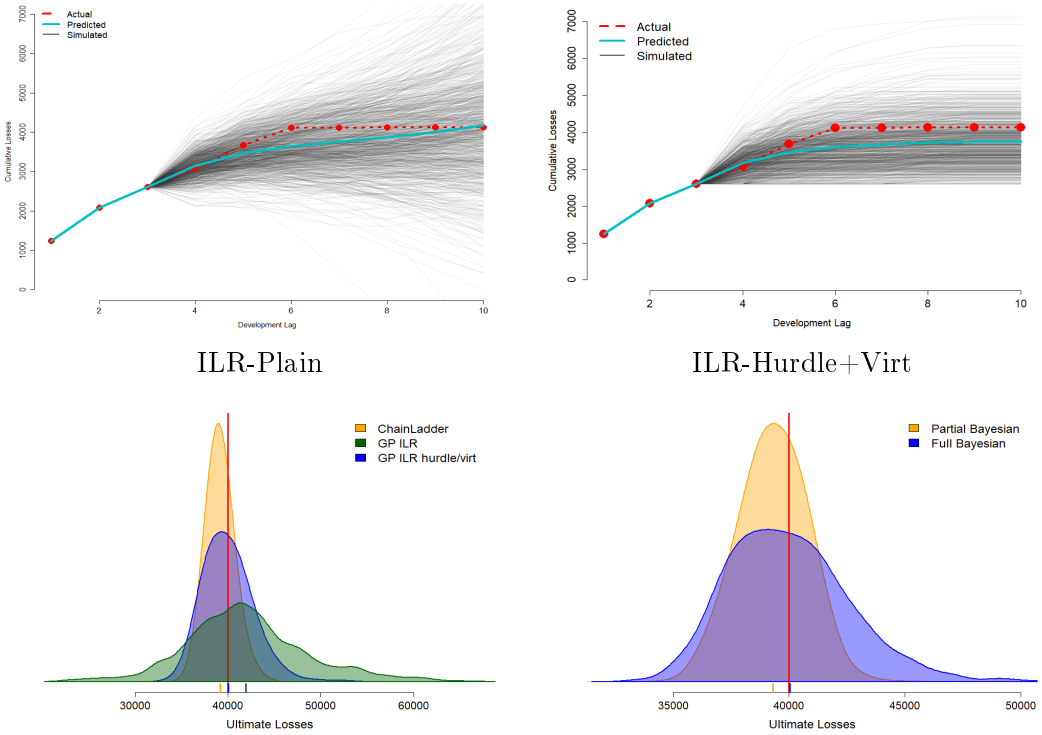

Figure 7: Top row: 1000 conditional simulations of future cumulative losses $CC_{p,q}$ for AY $= 1995, q = 1, \ldots, 10$ and a representative `comauto` triangle. The solid cyan line is the predictive mean of $CC_{p,q}$ and the dashed red line are the actual realized losses. (*Left:* ILR-Plain model; *Right:* ILR-Hurdle+Virt model) Bottom row: predictive density of $R_{ult} = \sum_p CC_{p,Q}$, together with the realized ultimate losses (vertical line) for the same triangle. *Left:* GP ILR-Plain model; *Right:* Partial-Bayesian vs Full-Bayesian for ILR-Hurdle+Virt.

smoothing mechanism that offers a recipe for extracting a latent ILR surface from a given loss triangle. This smoothing is very similar to kernel regression, prescribing a translation from $\mathcal{D}$ to $f_{p,q}$, and in fact resembles the Chain Ladder recipe that represents LDFs as a weighted average of observed empirical YoY development factors. However, unlike (2.4), the partial Bayesian model then sets "$s(x_*) := 0$" so that $f_*(x_*) \equiv m_*(x_*)$. Thus, there is no posterior variance of the latent f and the inference step returns a single "best" point estimate of the ILR surface, omitting model risk.

While the "middle" layer of uncertainty is now turned off, we maintain correlation uncertainty, continuing to employ MCMC to learn the spatial patterns in $f_{p,q}$ as encoded in ϑ. Thus, we keep residual uncertainty quantification by integrating, via MCMC sampling, over potential $m_*(x)$, to take into account that the MLE-based forecast ignores intrinsic correlation risk. Moreover, for forecast purposes, recognizing that future development is uncertain, we still inject randomness by generating realized loss quantities $Y(x)$ based on $m_*(x)$ and the hurdle model likelihood (3.1).

By construction, the partial Bayesian model will have narrower predictive bands for $Y_{p,q}$; furthermore those bands would be primarily based on the observation variance σ_q^2

rather than on the forecasting distance between the prediction cell and the upper triangle. The bottom-right panel of Figure 7 visualizes this effect by comparing the predictive distribution of R_{ult} from a ILR-Hurdle model relative to partial Bayesian version of the same. We observe that the predictive variance of a partial-Bayesian GP model is broadly similar to that of Mack CL. Since the latter tends to under-estimate predictive variance, we conclude that model uncertainty is a critical piece of the uncertainty quantification puzzle.

4.5 Handling multiple triangles

A single triangle yields a very small training dataset of just 50 or so observations. As a result, there is significant model risk, i.e. large uncertainty on how to complete it. This is indeed what we observe, with rather large predictive credible intervals for ultimate losses $CC_{p,Q}$. Note that this effect is in contrast to much existing literature where the predictive uncertainty is under-estimated. In turn, the natural remedy is to raise credibility by borrowing information from other triangles. This credibility boost can be quantified through both tighter hyperparameter posteriors, i.e. lower correlation risk and lower $s(x)$, i.e. lower model risk.

To borrow strength from multiple company data, we augment the original input data with a categorical covariate that is the indicator for the underlying company. Data from multiple companies are combined to form a single training set, with inputs now being trivariate, namely $x^i = (\text{Company}^i, \texttt{AY}^i, \texttt{DL}^i)$. Then the multiple-triangle squared-exponential kernel $C^{(SqExpMlt)}$ is defined as

$$\begin{aligned} C^{(SqExpMlt)}(x^i, x^j) :=& \eta^2 \exp(-\rho_{AY}^{-2}(\texttt{AY}^i - \texttt{AY}^j)^2 - \rho_{DL}^{-2}(\texttt{DL}^i - \texttt{DL}^j)^2) \ \cdot e^{-\rho_{Co}\cdot(1-\delta_{ij})}, \\ \text{where} \qquad \delta_{ij} =& \begin{cases} 0, & \text{Company}^i \neq \text{Company}^j \\ 1, & \text{otherwise} \end{cases} \qquad \text{and } \rho_{Co} > 0. \end{aligned} \tag{4.3}$$

In effect, the covariance between two data points is "discounted" by $e^{-\rho_{Co}}$ when the inputs come from different companies. A low value for ρ_{Co} implies that statistical gains are available from grouping company data and a high value implies the companies are relatively independent. In our approach we use the same η, ρ and σ_q parameters across grouped companies (which, in itself, stabilizes the analysis), although a more sophisticated, hierarchical approach for these parameters could also be adopted. By using this method, learning of Model Factors from one company can be influenced by such Model Factors from other companies. As before, we add linear kernels $C^{(lin)}$ from (2.7) to $C^{(SqExpMlt)}$.

An alternative solution is to build a categorical-input GP with a different ρ-correlation coefficient for each company pair that adjusts the squared exponential kernel to:

$$\tilde{C}^{(SqExpMlt)}(x^i, x^j) = \eta^2 \exp\Big(-\rho_{AY}^{-2}(\texttt{AY}^i - \texttt{AY}^j)^2 - \rho_{DL}^{-2}(\texttt{DL}^i - \texttt{DL}^j)^2 - \rho_{Co_i,Co_j}\Big),$$

where ρ_{Co_i,Co_j} measures the "distance" between Company^i and Company^j. Thus, $\tilde{C}(\cdot,\cdot)$ is the squared-exponential kernel on the trivariate input cells x where the first coordinate is interpreted as a categorical factor and has distance (interpreted as cross-triangle

correlation) $e^{-\rho_{Co_i,Co_j}}$ between factor levels Co_i, Co_j. Note that this specification leads to $n(n-1)/2$ hyper-parameters $\rho_{Co,i,j}$ to be estimated for a set of n triangles, limiting scability as n grows.

Running the multi-company model on a set of five `wkcomp` triangles (picked randomly from the dataset) we obtain the following estimates of ρ_{Co_i,Co_j}:

$$R^{(Co)} := \begin{pmatrix} 1 & & & & \\ 0.9553 & 1 & & & \\ 0.8587 & 0.8497 & 1 & & \\ 0.7467 & 0.7389 & 0.6641 & 1 & \\ 0.9498 & 0.9398 & 0.8449 & 0.7346 & 1 \end{pmatrix}. \tag{4.4}$$

Thus, we find that three companies labeled as "1", "2" and "5", form a "cluster" with high correlation among their ILRs, but Companies "3" and "4" are much less correlated. In comparison, implementing (4.3) to infer a single correlation hyperparameter ρ_{Co} yields the estimate $\rho_{Co} = 0.07345$, which is roughly the average of the above ρ_{Co_i,Co_j}'s. An important take-away is that the correlation structure is heterogeneous, so that assuming a constant cross-triangle correlation as in (4.3) is inappropriate, at least for randomly selected triangles.

A different approach to borrow strength from other run-off triangles was investigated in Shi and Hartman (2016), who imposed a hierarchial multivariate normal prior on the LDFs. Thus, development factors from different triangles are empirically correlated by assuming they are all drawn from the common $\mathcal{N}(\mu, \theta_{co})$ hyper-prior. Shi & Hartman postulate a known θ_{co} interpreted as the shrinkage parameter: as $\theta_{co} \to 0$, the LDFs across triangles will all collapse to the same value, effectively boosting credibility. In contrast to above, in our analysis we organically determine the between-triangle correlation in ILRs, that is learned like other hyper-parameters. We refer to Avanzi et al. (2016), Shi (2017), Shi et al. (2012) for further multi-triangle approaches, in particular relying on copula tools.

5 Conclusion and Outlook

Gaussian process models are a powerful centerpiece in the modern predictive analytics/data science toolkit. We demonstrate that they are also highly relevant for actuaries in the context of modeling loss development. The GP approach brings a rigorous Bayesian perspective that is simultaneously data-driven and fully stochastic, enabling precise, nonparametric quantification of both extrinsic and intrinsic sources of uncertainty. Our empirical work with the large-scale NAIC database demonstrate the GP-based models are able to properly account for the predictive distribution of cumulative claims and loss ratios. While our methods do not necessarily bring material improvement in raw accuracy (leading to similar RMSE measures relative to classical CL approaches), they do significantly improve coverage and percentile rank scores.

As a modeling tool, a major advantage of GPs is that they offer a wide latitude for customization. Above, we illustrated this strength by constructing a special Gaussian hurdle

likelihood to match the observed behavior of ILRs. Two other tailorings involved imposing a non-decreasing, yet non-parametric structure on the observation variances σ_q^2 and adding virtual observations. There remain several other directions for further investigation that we now briefly discuss.

Advanced Kernel Modeling: It would be worthwhile to investigate the realized dependence structure in the run-off triangles. Beyond the common "shape" in how claims cumulate over development years (i.e. a common set of ILRs), and the business trend over accident year, further dependence is likely. For example, operational changes in the claims management and business practice, reserving practice and legislative changes also take place and can potentially introduce calendar-year effects. Within the run-off triangle these could manifest themselves in a "diagonal dependence" for the ILRs, namely over ILRs for which $p+q=YR$ for a given YR. This also includes incorporating other trends through alternative linear (or polynomial) kernel specifications. One should also investigate alternative kernel families, such as the Matern family instead of the squared-exponential family (2.6) we employed, or alternative compound kernels.

Modeling Factors: Our analysis suggests that incremental loss ratios is the best object to model. In particular, this stems from the resulting additive structure in converting ILRs to cumulative claims which matches better the additive Gaussian structure intrinsic to GPs. Nevertheless, we expect that successful GP models could also be developed for LDFs or for cumulative loss ratios. As mentioned, Lally and Hartman (2018) achieved good performance with a GP model for $CC_{p,q}$. Full comparison of these competing frameworks remains to be done. In a related vein, additional analysis is warranted for incurred losses that could be adjusted up/down, in contrast to the paid losses discussed herein that are generally monotone.

Multiple Triangles: Finally, while we showed that GPs can be straightforwardly adapted to handle multi-triangle development, full investigation into this topic is beyond the scope of this paper. Multiple triangles raise two issues:

The first is computational tractability. Going beyond 10-20 triangles leads to consideration of many hundreds of cells and requires significant adjustments from off-the-shelf GP tools as full Bayesian MCMC with Stan becomes prohibitively expensive. One promising idea is to take advantage of the gridded nature of loss triangles. Such gridded structure implies that the GP covariance matrix C has a special "Kronecker" shape. The Kronecker Product (Flaxman et al.) approach decomposes a grid of input data into two sets of covariance functions allowing for a dramatic speedup in the inference models through faster matrix inversion formulas. The use of GPUs in performing GP calculations is also starting to be implemented in Stan and other probabilistic languages (e.g. Greta), and preliminary testing suggests that much larger models can be computationally tractable.

The second issue concerns the fundamental question of how to construct a relevant training dataset. For example, the full NAIC dataset includes more than 300 triangles and would still not be computationally feasible to consider all at once unless a supercomputer is available. Therefore, an appropriate *subset* should be identified given a concrete modeling task. This suggests to cluster the triangles prior to employing a multi-triangle model. This

can be done manually by the analyst who may identify which companies are likely, a priori, to be viewed as similar, or through a clustering algorithm. Another option would be to follow a two-step hierarchical procedure where single-triangle models are used to identify groupings and then multi-triangle analysis is conducted. Our initial investigations suggest that deciding what to do depends on the aims, e.g. boosting credibility of a given triangle forecasting, or identifying common trends across triangles (i.e. inference of the covariance structure), or capturing cross-triangle correlation for risk allocation purposes.

Acknowledgment

We thank the anonymous reviewers for many thoughtful suggestions that improved the original manuscript. Support from CAS Committee on Knowledge Extension Research is gratefully acknowledged.

References

B. Ankenman, B. L. Nelson, and J. Staum. Stochastic kriging for simulation metamodeling. *Operations Research*, 58(2):371–382, 2010.

B. Avanzi, G. Taylor, P. A. Vu, and B. Wong. Stochastic loss reserving with dependence: A flexible multivariate Tweedie approach. *Insurance: Mathematics and Economics*, 71: 63–78, 2016.

M. Betancourt. Robust Gaussian processes in Stan, 2017. Online post at `https://betanalpha.github.io/assets/case_studies/gp_part1/part1.html`.

P.-C. Bürkner, J. Gabry, and A. Vehtari. Approximate leave-future-out cross-validation for time series models. *Journal of Statistical Computation and Simulation*, pages 1–25, 2020.

B. Carpenter, A. Gelman, M. D. Hoffman, D. Lee, B. Goodrich, M. Betancourt, M. Brubaker, J. Guo, P. Li, and A. Riddell. Stan: A probabilistic programming language. *Journal of statistical software*, 76(1), 2017.

L. Carvalho. An improved evaluation of Kolmogorov's distribution. *Journal of Statistical Software*, 65(3):1–7, 2015.

D. R. Clark. LDF curve-fitting and stochastic reserving: a maximum likelihood approach. CAS Reserves Call Paper Program, 2003.

D. Duvenaud, J. R. Lloyd, R. Grosse, J. B. Tenenbaum, and Z. Ghahramani. Structure discovery in nonparametric regression through compositional kernel search. In *Proceedings of the 30th International Conference on Machine Learning*, pages 1166–1174, 2013.

P. D. England and R. J. Verrall. Stochastic claims reserving in general insurance. *British Actuarial Journal*, 8(3):443–518, 2002.

P. D. England, R. J. Verrall, and M. V. Wüthrich. Bayesian over-dispersed Poisson model and the Bornhuetter & Ferguson claims reserving method. *Annals of Actuarial Science*, 6(2):258–283, 2012.

S. Flaxman, A. Gelman, D. Neill, A. Smola, A. Vehtari, and A. G. Wilson. Fast hierarchical Gaussian processes. Technical report, Preprint at http://sethrf.com/files/fast-hierarchical-GPs.pdf.

A. Forrester, A. Sobester, and A. Keane. *Engineering design via surrogate modelling: a practical guide*. John Wiley & Sons, 2008.

A. Gangopadhyay and W.-C. Gau. Credibility modeling via spline nonparametric regression. In *The Casualty Actuarial Society Forum Winter 2003*, page 215. Citeseer, 2003.

M. Gesmann, D. Murphy, Y. W. Zhang, A. Carrato, M. Wüthrich, F. Concina, and E. Dal Moro. *ChainLadder: Statistical Methods and Models for Claims Reserving in General Insurance*, 2018. R package version 0.2.9.

T. Gneiting and A. E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

A. Jordan, F. Krüger, and S. Lerch. Evaluating probabilistic forecasts with scoringRules. *Journal of Statistical Software*, 90(12), 2019.

K. Kuo. DeepTriangle: A deep learning approach to loss reserving. *Risks*, 7(3):97, 2019.

N. Lally and B. Hartman. Estimating loss reserves using hierarchical bayesian gaussian process regression with input warping. *Insurance: Mathematics and Economics*, 82: 124–140, September 2018.

T. Mack. Distribution-free calculation of the standard error of chain ladder reserve estimates. *ASTIN Bulletin: The Journal of the IAA*, 23(2):213–225, 1993.

T. Mack. Which stochastic model is underlying the chain ladder method? *Insurance: mathematics and economics*, 15(2-3):133–138, 1994.

M. Merz and M. V. Wüthrich. Paid–incurred chain claims reserving method. *Insurance: Mathematics and Economics*, 46(3):568–579, 2010.

G. Meyers. Stochastic loss reserving using Bayesian MCMC models. Technical report, Casualty Actuarial Society New York, 2015.

G. Meyers and P. Shi. Loss reserving data pulled from NAIC Schedule P. Casualty Actuarial Society Website `https://www.casact.org/research/index.cfm?fa=loss_reserves_data`, year = 2011,.

G. G. Meyers and P. Shi. The retrospective testing of stochastic loss reserve models. In *Casualty Actuarial Society E-Forum, Summer*, 2011.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

O. Roustant, D. Ginsbourger, and Y. Deville. DiceKriging, DiceOptim: Two R packages for the analysis of computer experiments by kriging-based metamodeling and optimization. *Journal of Statistical Software*, 51(1):1–55, 2012.

P. Shi. A multivariate analysis of intercompany loss triangles. *Journal of Risk and Insurance*, 84(2):717–737, 2017.

P. Shi and B. M. Hartman. Credibility in loss reserving. *North American Actuarial Journal*, 20(2):114–132, 2016.

P. Shi, S. Basu, and G. G. Meyers. A Bayesian log-normal model for multivariate loss reserving. *North American Actuarial Journal*, 16(1):29–51, 2012.

G. A. Spedicato, G. P. Clemente, and F. Schewe. The use of GAMLSS in assessing the distribution of unpaid claims reserves. In *Casualty Actuarial Society E-Forum, Summer 2014-Volume 2*, 2014.

G. Taylor. Bayesian chain ladder models. *ASTIN Bulletin: The Journal of the IAA*, 45 (1):75–99, 2015.

M. V. Wüthrich. Neural networks applied to chain-ladder reserving. *European Actuarial Journal*, 8(2):407–436, 2018.

M. V. Wüthrich and M. Merz. *Stochastic claims reserving methods in insurance*, volume 435. John Wiley & Sons, 2008.

Y. Zhang and V. Dukic. Predicting multivariate insurance loss payments under the Bayesian copula framework. *Journal of Risk and Insurance*, 80(4):891–919, 2013.

Y. Zhang, V. Dukic, and J. Guszcza. A Bayesian non-linear model for forecasting insurance loss payments. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 175(2):637–656, 2012.

A Results for all business lines

Algorithm	Claims RMSE	LR RMSE	Coverage	CRPS	NLPD	K-S
	`comauto` : $z_{KS} = 0.1461$					
Mack CL	8718	0.057	0.774	217821	1304.2	0.184
Bootstrap CL	8820	0.122	0.774	215647	1306.3	0.172
ILR-Plain	23947	0.141	0.941	541374	1464.7	0.244
ILR Hurdle	10442	0.146	0.929	276313	1373.4	0.174
Hurdle+Virt	10211	0.146	0.893	246961	1369.6	0.202
	`wkcomp` : $z_{KS} = 0.1767$					
Mack CL	24726	0.049	0.509	388621	1305.6	0.306
Bootstrap CL	24895	0.052	0.544	389144	1232.5	0.292
ILR-Plain	84728	0.137	0.983	1150989	1113.5	0.244
ILR Hurdle	38638	0.081	0.912	640873	1072.1	*0.126*
Hurdle+Virt	39577	0.079	0.895	604892	1072.4	*0.142*
	`medmal` : $z_{KS} = 0.3754$					
Mack CL	99896	0.151	0.500	423580	265.9	*0.274*
Bootstrap CL	102084	0.153	0.667	400111	270.3	*0.238*
ILR-Plain	60527	0.224	0.917	332687	261.2	0.440
ILR Hurdle	31319	0.123	0.917	164267	249.5	*0.328*
Hurdle+Virt	24987	0.116	0.917	144724	246.5	*0.279*
	`othliab` : $z_{KS} = 0.1368$					
Mack CL	46078	0.062	0.813	448176	1320.6	0.166
Bootstrap CL	46824	0.068	0.792	456674	1396.8	0.156
ILR-Plain	84395	0.124	0.938	788754	1495.2	0.162
ILR Hurdle	47202	0.131	0.938	396892	1394.9	*0.082*
Hurdle+Virt	31890	0.124	0.896	309136	1390.1	0.187
	`ppauto` : $z_{KS} = 0.1436$					
Mack CL	115269	0.048	0.667	1269875	1526.5	0.376
Bootstrap CL	118070	0.052	0.609	1355631	1539.1	0.369
ILR-Plain	113911	0.177	0.977	2577475	1673.9	0.231
ILR Hurdle	356443	0.090	0.954	2042819	1574.0	0.294
Hurdle+Virt	65730	0.103	0.920	1202087	1557.2	0.357
	`prodliab` : $z_{KS} = 0.3614$					
Mack CL	49841	0.148	0.769	164597	220.4	*0.224*
Bootstrap CL	51709	0.162	0.769	182050	243.8	*0.137*
ILR-Plain	34317	0.144	0.692	152861	243.5	*0.283*
ILR Hurdle	47942	0.092	0.769	164736	232.4	*0.226*
Hurdle+Virt	40521	0.098	0.846	143834	223.3	*0.251*

B Step Ahead RMSE

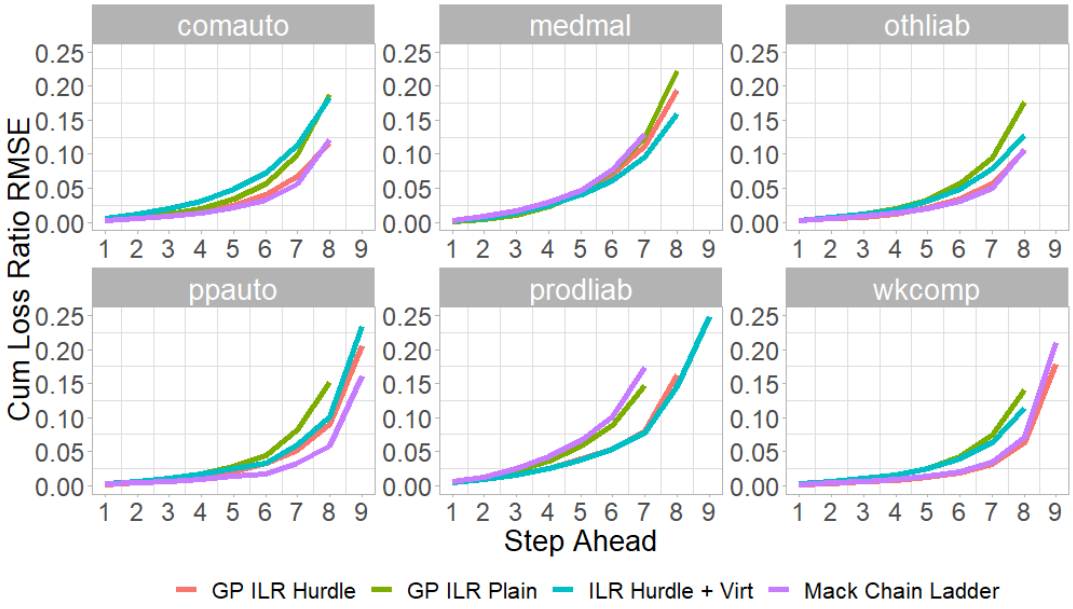

Figure 8: RMSE of cumulative loss ratios as a function of step-ahead n across the six business lines. Lags where RMSE exceeds 0.25 are clipped from the panels.

C Typical ILR Triangle

Table 2: Sample run-off triangle from `comauto` with Accident Years p in rows and Development Lags q in columns. Top: cumulative losses $CC_{p,q}$; bottom: incremental loss ratios $L_{p,q}$.

	1	2	3	4	5	6	7	8	9	10
1988	952	1529	2813	3647	3724	3832	3899	3907	3911	3912
1989	849	1564	2202	2432	2468	2487	2513	2526	2531	2527
1990	983	2211	2830	3832	4039	4065	4102	4155	4268	4274
1991	1657	2685	3169	3600	3900	4320	4332	4338	4341	4341
1992	932	1940	2626	3332	3368	3491	3531	3540	3540	3583
1993	1162	2402	2799	2996	3034	3042	3230	3238	3241	3268
1994	1478	2980	3945	4714	5462	5680	5682	5683	5684	5684
1995	1240	2080	2607	3080	3678	4116	4117	4125	4128	4128
1996	1326	2412	3367	3843	3965	4127	4133	4141	4142	4144
1997	1413	2683	3173	3674	3805	4005	4020	4095	4132	4139

	Init	2	3	4	5	6	7	8	9	10
1988	0.164	0.099	0.221	0.143	0.013	0.019	0.012	0.001	0.001	0.000
1989	0.173	0.146	0.130	0.047	0.007	0.004	0.005	0.003	0.001	-0.001
1990	0.180	0.225	0.113	0.184	0.038	0.005	0.007	0.010	0.021	0.001
1991	0.320	0.199	0.094	0.083	0.058	0.081	0.002	0.001	0.001	0.000
1992	0.179	0.193	0.132	0.135	0.007	0.024	0.008	0.002	0.000	0.008
1993	0.222	0.237	0.076	0.038	0.007	0.002	0.036	0.002	0.001	0.005
1994	0.296	0.301	0.193	0.154	0.150	0.044	0.000	0.000	0.000	0.000
1995	0.227	0.154	0.096	0.087	0.109	0.080	0.000	0.001	0.001	0.000
1996	0.254	0.208	0.183	0.091	0.023	0.031	0.001	0.002	0.000	0.000
1997	0.285	0.256	0.099	0.101	0.026	0.040	0.003	0.015	0.007	0.001