DETRA NOTE

DETRA NOTE 2025-2

INSURANCE RISK CLASSIFICATION WITH GENERALIZED GAUSSIAN PROCESS REGRESSION MODELS

Donatien Hainaut & Michel Denuit

Detralytics

DISCLAIMER

The content of the Detra Notes for a pedagogical use only. Each business case is so specific that a careful analysis of the situation is needed before implementing a possible solution. Therefore, Detralytics does not accept any liability for any commercial use of the present document. Of course, the entire team remain available if the techniques presented in this Detra Note required your attention.

Detralytics
Avenue du Boulevard 21 Box 5 - 1210 Bruxelles (Belgium)
1-7 Cour Valmy, Le Belvédère - 92800 Puteaux (France)
www.detralytics.com - info@detralytics.eu

ABSTRACT

This paper proposes a new approach to risk classification based on Generalized Gaussian Process Regression (GGPR). The response under consideration obeys a distribution belonging to the Exponential Dispersion (ED) family. It typically corresponds to a claim count or a claim severity in the context of insurance studies. GGPR is a supervised machine learning method with Bayesian flavor. Individual random effects obeying a multivariate Normal distribution are connected with the help of their covariance matrix built from a so-called kernel function. The latter enforces smoothness, borrowing information from similar risk profiles. Bayesian Generalized Linear Models (GLMs) and Generalized Additive Models (GAMs) are recovered as special cases, assuming a highly-structured prior covariance matrix. Compared to the existing literature, this paper innovates to account for the specificity of data entering insurance studies. First, proper risk exposures are included in model formulation and development. Second, parameters are estimated by minimizing deviance instead of an approximated log-likelihood. Third, categorical features that are often encountered in insurance data bases are coded with the help of an embedding method based on Burt matrices. Fourth, K-means clustering is used to reduce the dimension of the problem and create model points within large insurance portfolios. Numerical illustrations performed on publicly available insurance data sets illustrate the relevance of the GGPR approach to risk classification. Benchmarked against the classical GLM, the performances of GGPR turn out to be excellent given its reduced number of parameters. This suggests that GGPR nicely enriches the actuarial toolkit by providing preliminary predictions that can then be structured with additive scores like those entering GLMs and GAMs.

Keywords: Exponential Dispersion family, Mixed models, Risk classification, Categorical embedding, Burt distance, Model points.

1 Introduction and motivation

Bayesian ideas and techniques entered the actuarial toolkit in the late 1960s with the pioneering works by Bühlmann and Straub devoted to empirical Bayes credibility techniques; see e.g. Makov (2002) for a review. Boskov and Verrall (1994) demonstrated the interest of latent variables to smooth spatial effects in motor insurance, resorting to multivariate Normal prior distribution with a spatially structured covariance matrix. Dimakos and Rattalma (2002) extended this idea by proposing a Bayesian version of the Generalized Linear Model (GLM) approach to insurance risk classification. Denuit and Lang (2004) further extended this approach to Generalized Additive Models (GAMs) under the Bayesian structured additive regression setting. We refer to Denuit and Lang (2004) and Klein et al. (2014) for more information and for insurance case studies.

However, the Bayesian structured additive regression approach to insurance pricing is in essence parametric: the score is structured in an additive way, resulting in a covariance matrix with different blocks where smoothness is induced for continuous and spatial features. This is in contrast with the Generalized Gaussian Process Regression (GGPR) proposed by Chan and Dong (2011) where only similarity between risk profiles matters. GGPR applies to response obeying a distribution in the Exponential Dispersion (ED) family, exactly as with GLMs and GAMs. It is a non-parametric kernel-based machine learning approach with Bayesian flavor. The GGPR approach complements Bayesian structured additive regression by allowing the actuary to derive a preliminary risk assessment solely based on the similarity between risk profiles in the portfolio. Approximate inference is generally needed because the Gaussian prior and the ED distribution of the response are not conjugated. This is performed with the help of Laplace approximation exhibiting good performances on large data sets.

The covariance matrix entering the Gaussian prior in GGPR is obtained from a so-called kernel. The latter is a symmetric positive definite function which encodes the degree of similarity between any two vectors of features. The idea is that similarity is strong when they are close in the feature space and weak when they are far away from each other. Kernel functions play a central role in GGPR and hyper-parameters entering kernels need to be selected with great care.

This paper introduces GGPR to the actuarial community and adapts this machine learning tool to the specificity of insurance studies. From a methodological point of view, the contributions of this paper are as follows:

1. Compared to Chan and Dong (2011), GGPR is adapted to account for contracts with different exposures (duration for claim counts or claim numbers for severities, for instance). This is important for actuarial applications where exposures often play a major role.

2. Furthermore, instead of using an approximation of the log-likelihood to fit the GGPR model, we minimize the exact deviance which is the standard goodness-of-fit measure for actuarial models.

3. Also, classical GGPRs cannot directly deal with categorical features. This is because covariance kernels require each feature to be associated with a distance metric while cat-

egorical variables by definition lack such measures. This issue must be addressed since categorical features generally represent a large fraction of the information recorded in insurance data bases. In this paper, we propose an embedding technique based on a contingency table. Categorical features are included in the analysis by using Burt's distance to assess proximity, following Hainaut (2019) and Jamotton et al. (2024). This method is totally transparent and founded on a solid theoretical background. It offers an alternative to neural networks with categorical embedding layers where low-dimensional features extracted from the neural network translate categorical information into numerical representations. See, e.g., Shi and Shi (2023), Avanzi et al. (2024), Carlin and Benjamini (2025), and Wang et al. (2025).

4. Our last contribution concerns the analysis of large data sets. One limitation of GGPRs is the computation complexity for a large sample size. Due to computational constraints, GGPR does not scale well with large data sets like insurance databases which often range in the hundreds of thousands of records. Various methods, such as sparse approximations and inducing points, have been proposed to address this issue, but they often involve trade-offs between accuracy and complexity. In this paper, we use a reduction dimension based on a K-means clustering algorithm. The latter converts the initial data set into a limited number of "model points" which may be seen as standard dominant risk profiles in the portfolio. This approach can be seen as a pragmatic actuarial counterpart to Nearest Neighbor Gaussian Process (NNGP) using conditional independence given information from neighboring points for large data sets, resulting in sparse precision matrices.

Numerical illustrations based on two publicly available insurance data sets demonstrate the relevance of the proposed approach in the analysis of insurance claim frequencies and severities compared to standard GLM methods.

The outline of the paper is as follows. Section 2 recaps the GGPR approach to responses obeying a distribution in the ED family, allowing for different risk exposures. Section 3 discusses the calibration of the parameters entering kernel functions by minimizing the deviance. Section 4 develops an embedding method for categorical features, compatible with GGPR. Section 5 proposes a solution for managing large data sets based on a batch K-means algorithm. Section 6 applies the proposed approach to a couple of publicly available insurance data sets. The first one concerns motor third-party liability insurance in France, while the second one involves motorcycle insurance in Sweden. The final Section 7 summarizes the main findings of the paper and discusses remaining issues. The proofs of the technical results as well as supplementary material are gathered in appendix.

2 Generalized Gaussian process regression

2.1 Responses and key ratios

In risk classification studies, actuaries generally aim to estimate expected claim frequencies and severities in order to derive the amount of pure premium according to risk profile. The latter is summarized in a vector of features representing the information available to the insurer. Sometimes, the probability of an event is the quantity of interest, like the no-claim probability for instance. Every insurance study must account for a proper exposure to risk reflecting the “volume” of the risk (like coverage duration, for instance).

The available information is gathered in a vector $\boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^m$ of features $x_1, x_2, \ldots, x_m$. These features are used to explain the behavior of a random continuous or discrete response R accounting for the corresponding risk exposure denoted by ω. For instance, R can be the total amount of claims, the total number of claims or of event occurrences (like defaults) for a group of ω insurance policies with common features $\boldsymbol{x}$.

Following Ohlsson and Johansson (2010), let us define the key ratio Y as the response R divided by the corresponding exposure ω, that is, $Y = \frac{R}{\omega}$. The domain of Y is denoted as $\mathcal{Y} \subset \mathbb{R}$. If R and Y are continuous random variables then their respective probability density functions $p_R(r \,|\, \boldsymbol{x})$ and $p(y \,|\, \boldsymbol{x})$ are related by $p(y \,|\, \boldsymbol{x}) = \omega p_R(\omega y \,|\, \boldsymbol{x})$. If R and Y are discrete random variables then their respective probability mass functions $p_R(r \,|\, \boldsymbol{x})$ and $p(y \,|\, \boldsymbol{x})$ satisfy $p(y \,|\, \boldsymbol{x}) = p_R(\omega y \,|\, \boldsymbol{x})$.

Throughout this paper, we assume that the distribution of Y belongs to the Exponential Dispersion (ED) family of distributions. This means that the probability density or mass function of Y is of the form

$$p\left(y \,|\, \boldsymbol{x}\right) = \exp\left\{\frac{y\,\theta(\boldsymbol{x}) - \gamma\left(\theta(\boldsymbol{x})\right)}{\phi/\omega} + c(y, \phi, \omega)\right\}, \;\; y \in \mathcal{Y}, \tag{2.1}$$

where the canonical parameter is a function $\theta(\boldsymbol{x}) : \mathcal{X} \to \mathbb{R}$ of the available features and where $\phi \in \mathbb{R}^+$ is a dispersion parameter. The function $\gamma(\cdot)$ in (2.1) is $\mathcal{C}^2$ and admits an inverse second-order derivative whereas $\theta(\boldsymbol{x})$ does not enter the normalizing function $c(\cdot)$. Table 2.1 summarizes the most useful distributions of key ratios for actuarial purposes. The corresponding functions $\theta(\cdot)$, $\gamma(\cdot)$ and parameter ϕ entering representation (2.1) are provided in Table 2.2. We refer the reader e.g. to Ohlsson and Johansson (2010), Denuit et al. (2019) or Wüthrich and Merz (2023) for a textbook treatment of the ED distributions and their applications to insurance.

Response, R	Key ratio, $Y = R/\omega$	$p\left(y \,|\, \boldsymbol{x}\right)$
Sum of ω Normal claim sizes $\mathcal{N}(\omega\mu(\boldsymbol{x}), \omega\sigma^2)$	Y is Normal, $\mathcal{N}(\mu(\boldsymbol{x}), \frac{\sigma^2}{\omega})$	$\frac{\sqrt{\omega}}{\sigma\sqrt{2\pi}} e^{-\frac{\omega}{2}\left(\frac{y-\mu(\boldsymbol{x})}{\sigma}\right)^2}$
Sum of ω Gamma claim sizes $\mathcal{G}\left(\omega\alpha, \beta(\boldsymbol{x})\right)$	Y is Gamma, $\mathcal{G}\left(\omega\alpha, \omega\beta(\boldsymbol{x})\right)$	$\frac{(\omega\beta(\boldsymbol{x}))^{\omega\alpha}}{\Gamma(\omega\alpha)} y^{\omega\alpha-1} e^{-\omega\beta(\boldsymbol{x})y}$
Sum of ω Inverse Gaussian claim sizes $\mathcal{IG}\left(\omega\mu(\boldsymbol{x}), \omega^2\alpha\right)$	Y is Inverse Gaussian, $\mathcal{IG}\left(\mu(\boldsymbol{x}), \omega\alpha\right)$	$\sqrt{\frac{\alpha\omega}{2\pi y^3}} e^{-\frac{\alpha\omega(y-\mu(\boldsymbol{x}))^2}{2y\mu(\boldsymbol{x})^2}}$
Sum of ω Poisson claim counts $\mathcal{P}\left(\lambda(\boldsymbol{x})\,\omega\right)$	ωY is Poisson, $\mathcal{P}\left(\lambda(\boldsymbol{x})\,\omega\right)$	$e^{-\omega\lambda(\boldsymbol{x})} \frac{(\omega\lambda(\boldsymbol{x}))^{\omega y}}{(\omega y)!}$
Sum of ω Bernoulli event occurrences $\mathcal{B}\left(\omega, p(x)\right)$	ωY is Binomial, $\mathcal{B}\left(\omega, p(x)\right)$	$\left(\begin{array}{c}\omega \\ \omega y\end{array}\right) p(\boldsymbol{x})^{\omega y}\left(1 - p(\boldsymbol{x})\right)^{\omega(1-y)}$

Table 2.1: Most common statistical distributions used for actuarial analysis.

Key ratio, $Y = R/\omega$	$\theta(\boldsymbol{x})$	ϕ	$\gamma(z)$	$\mathbb{E}(Y)$	$\mathbb{V}(Y)$	$v(z)$
$\mathcal{N}(\mu(\boldsymbol{x}), \frac{\sigma^2}{\omega})$	$\mu(\boldsymbol{x})$	σ^2	$z^2/2$	$\mu(\boldsymbol{x})$	$\frac{\sigma^2}{\omega}$	1
$\mathcal{G}(\omega\alpha, \omega\beta(\boldsymbol{x}))$	$-\frac{\beta(\boldsymbol{x})}{\alpha}$	$\frac{1}{\alpha}$	$-\ln(-z)$	$\frac{\alpha}{\beta(\boldsymbol{x})}$	$\frac{\alpha}{\omega\beta(\boldsymbol{x})^2}$	z^2
$\mathcal{IG}(\mu(\boldsymbol{x}), \omega\alpha)$	$-\frac{1}{2\mu(\boldsymbol{x})^2}$	$\frac{1}{\alpha}$	$-\sqrt{-2z}$	$\mu(\boldsymbol{x})$	$\frac{\mu(\boldsymbol{x})^3}{\omega\alpha}$	z^3
$\mathcal{P}(\lambda(\boldsymbol{x})\,\omega)$	$\ln\lambda(\boldsymbol{x})$	1	e^z	$\lambda(\boldsymbol{x})$	$\frac{\lambda(\boldsymbol{x})}{\omega}$	z
$\mathcal{B}(\omega, p(x))$	$\ln\frac{p(\boldsymbol{x})}{1-p(\boldsymbol{x})}$	1	$\ln(1+e^z)$	$p(\boldsymbol{x})$	$\frac{p(\boldsymbol{x})(1-p(\boldsymbol{x}))}{\omega}$	$z(1-z)$

Table 2.2: Reformulation of distributions in Table 2.1 as ED distributions with corresponding first moments and variance functions.

2.2 Mean-variance structure

For a given risk profile $\boldsymbol{x}$, the conditional mean and variance of Y are respectively given by

$$\mu(\boldsymbol{x}) = \mathbb{E}(Y|\boldsymbol{x}) = \gamma'(\theta(\boldsymbol{x})) \text{ and } \sigma^2(\boldsymbol{x}) = \mathbb{V}(Y|\boldsymbol{x}) = \frac{\phi}{\omega}\gamma''(\theta(\boldsymbol{x}))\,.$$

We deduce that $\theta(\boldsymbol{x})$ is related to $\mu(\boldsymbol{x})$ by

$$\theta(\boldsymbol{x}) \quad = \quad \gamma'^{-1}(\mu(\boldsymbol{x}))\,,$$

while $\sigma^2(\boldsymbol{x})$ depends on $\mu(\boldsymbol{x})$ through the variance function $v(.)$:

$$\sigma^2(\boldsymbol{x}) = \frac{\phi}{\omega}v(\mu(\boldsymbol{x})) \text{ where } v(\cdot) = \gamma''\left(\gamma'^{-1}(\cdot)\right)\,.$$

The variance functions of main ED distributions are provided in Table 2.2.

2.3 Link function

With Generalized Linear Models (GLMs), the conditional expectation $\mu(\boldsymbol{x})$ of Y is related to a linear combination of features $\boldsymbol{\beta}^\top\boldsymbol{x}$ through a monotonic link function selected by the analyst. The link function is henceforth denoted as $l(.)$ and the GLM regression specification is as follows:

$$l(\mu(\boldsymbol{x})) = \boldsymbol{\beta}^\top\boldsymbol{x}\,, \tag{2.2}$$

where $\boldsymbol{\beta} \in \mathbb{R}^m$ is a vector of regression coefficients to be estimated from claim data. GAMs replace the linear combination $\boldsymbol{\beta}^\top\boldsymbol{x}$ in (2.2) with a sum of smooth functions of continuous features, to be estimated from claim data.

If the link function is such that $l(\mu(\boldsymbol{x})) = \theta(\boldsymbol{x})$ then the link function is called canonical. Canonical link functions are given by $l(\cdot) = \gamma'^{-1}(\cdot)$. Table 2.3 lists the canonical link functions for distributions in Table 2.1. In certain circumstances, using a canonical link function simplifies developments. In practice, the canonical link is used by actuaries for Normal, Poisson, and Binomial responses whereas the power link is replaced by the logarithmic one for Gamma and Inverse Gaussian responses. This choice is made for two reasons. Firstly, with the (negative) power link function, we can generate a negative expected value for some combinations of features. Secondly, the power link introduces a singularity in the Gaussian process predictor.

Link function	$l(\mu) = \gamma'^{-1}(\mu)$	Canonical link for
Identity	μ	Normal
Power	$-\mu^{-1}$	Gamma
	$-\frac{1}{2}\mu^{-2}$	Inv. Gaussian
Log	$\ln \mu$	Poisson
Logit	$\ln \frac{\mu}{1-\mu}$	Binomial

Table 2.3: Canonical link functions.

2.4 GGPR model

In the framework of GGPR, we consider a noised version of (2.2). More precisely, we assume that

$$l\left(\mu(\boldsymbol{x})\right) = g(\boldsymbol{x}) + \epsilon\,, \tag{2.3}$$

where $\epsilon \sim \mathcal{N}(0, \sigma^{*2})$ and $g(\boldsymbol{x}) : \mathcal{X} \to \mathbb{R}$ is a Gaussian process. These two components are assumed to be independent and the responses are assumed to be conditionally independent, given the random effects in (2.3). From an actuarial point of view, the random effect ϵ captures residual heterogeneity and opens the door to credibility adjustments. In following developments, we will see that σ^{*2} is an hyperparameter of shrinkage tuning the numerical robustness of the GGPR. Notice that Rasmussen and Williaws (2006) have not allowed for a shrinkage parameter for classification in Binomial and Multinomial models.

A Gaussian process is a collection $\{g(\boldsymbol{x})\,,\,\boldsymbol{x} \in \mathcal{X}\}$ such that for any $n \in \mathbb{N}$ and any $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X}$, the random vector $(g(\boldsymbol{x}_1), \ldots, g(\boldsymbol{x}_n))$ obeys a joint multivariate Gaussian distribution. Without loss of generality, the mean vector is set to zero and the covariance matrix is defined by a kernel function $k(\boldsymbol{x}, \boldsymbol{x}')$ as

$$\mathbb{C}\left(g(\boldsymbol{x}), g(\boldsymbol{x}')\right) = k(\boldsymbol{x}, \boldsymbol{x}^{'}) \text{ for } (\boldsymbol{x}, \boldsymbol{x}^{'}) \in \mathcal{X} \times \mathcal{X}\,.$$

The kernel $k(\boldsymbol{x}, \boldsymbol{x}^{'})$ controls the correlation between the risk profiles $\boldsymbol{x}$ and $\boldsymbol{x}^{'}$, being large if these profiles are similar. A necessary and sufficient condition for the function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ to be a valid kernel is that the $n \times n$ matrix of $k(\boldsymbol{X}, \boldsymbol{X}) = \left(k(\boldsymbol{x}_i, \boldsymbol{x}_j)\right)_{i,j=1,\ldots,n}$ is positive semi-definite for all possible $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X}$. This means that $\boldsymbol{c}^\top k(\boldsymbol{X}, \boldsymbol{X})\boldsymbol{c} \geq 0$ holds for any $\boldsymbol{c} \in \mathbb{R}^n$ where $\boldsymbol{X} = \left(\boldsymbol{x}_i^\top\right)_{i=1,\ldots,n}$.

2.5 Kernel functions

The kernel function is an important ingredient of the GGPR method. This is because its choice dictates the structure of the covariance matrix of the Gaussian latent process and thus the way to assess proximity between risk profiles. Kernels are sometimes referred to as similarity functions because they measure how similar pairs of risk profiles are to each other.

Let $\|\cdot\|_2$ denote the Euclidean distance. Table 2.4 presents the kernels most often encountered in the literature, referred to as radial basis function (RBF), rational quadratic (RQ), Matern 3/2 (M32), Matern 5/2 (M52). These kernels are used in the numerical illustrations proposed in this paper.

The RBF kernel is often used to model covariance function for Gaussian processes. It enforces smoothness, with similarity between nearby risk profiles rapidly decaying as the Euclidean distance between them increases. RBF kernels are commonly used in regression and classification tasks where the relationship between the features and the prediction is expected to be non-linear and smooth. The RQ kernel is another popular choice for capturing non-linear relationships. It computes how similar two risk profiles are by raising the Euclidan distance to some power. The class of Matern kernels generalizes RBF with an additional parameter controlling the smoothness of the resulting function. Their name refers to the statistician Bertil Matérn who studied the spatial organization of forests and proposed several covariance functions which turned out to be useful in many applications beyond forestry. Table 2.4 considers two particular Matern kernels whose abbreviations refer to the particular value of the parameters.

Compared to GLM models which often involve many regression coefficients β_j to be estimated from claim data, GGPR has thus very few parameters (two or three for the kernels listed in Table 2.4) as it mainly relies on the similarities, measured by distance between policies, for estimating key ratios. Notice that kernels can be combined by multiplication to customize the covariance structure.

Kernel	$k(\boldsymbol{x}, \boldsymbol{x}')$	Hyper-parameter
RBF	$\sigma^2 \exp\left(-\frac{\lVert\boldsymbol{x}-\boldsymbol{x}'\rVert_2}{2\varphi^2}\right)$	$\boldsymbol{\Theta} = \{\sigma, \varphi \in \mathbb{R}^+\}$
RQ	$\sigma^2 \left(1 + \frac{\lVert\boldsymbol{x}-\boldsymbol{x}'\rVert_2^2}{2\,d\,\varphi^2}\right)^{-d}$	$\boldsymbol{\Theta} = \{\sigma, \varphi, d \in \mathbb{R}^+\}$
M32	$\sigma^2 \left(1 + \frac{\sqrt{3}\lVert\boldsymbol{x}-\boldsymbol{x}'\rVert_2}{\rho}\right) \exp\left(-\frac{\sqrt{3}\lVert\boldsymbol{x}-\boldsymbol{x}'\rVert_2}{\rho}\right)$	$\boldsymbol{\Theta} = \{\sigma, \varphi \in \mathbb{R}^+\}$
M52	$\sigma^2 \left(1 + \frac{\sqrt{5}\lVert\boldsymbol{x}-\boldsymbol{x}'\rVert_2}{\rho} + \frac{5\lVert\boldsymbol{x}-\boldsymbol{x}'\rVert_2^2}{3\rho^2}\right) \exp\left(-\frac{\sqrt{5}\lVert\boldsymbol{x}-\boldsymbol{x}'\rVert_2}{\rho}\right)$	$\boldsymbol{\Theta} = \{\sigma, \varphi \in \mathbb{R}^+\}$

Table 2.4: Common kernels and their hyper-parameters gathered in the vector $\boldsymbol{\Theta}$.

2.6 Laplace approximation

We consider a sample set $\mathcal{D}$ made of individual observations $(\boldsymbol{x}_i, y_i)$ for $i = 1, ...n$. We adopt the following notations: $\boldsymbol{y} = (y_i)_{i=1,\ldots,n}$, $\boldsymbol{X} = \left(\boldsymbol{x}_i^\top\right)_{i=1,\ldots,n}$, $\mathcal{D} = (\boldsymbol{X}, \boldsymbol{y})$, $g_i = g(\boldsymbol{x}_i) + \epsilon_i$ and $\boldsymbol{g} = (g_i)_{i=1,\ldots,n}$. From a Bayesian perspective, we encode our belief that instances of $l\left(\mu(\boldsymbol{x})\right)$ are drawn from a Gaussian process g with zero mean and covariance function $k(\boldsymbol{x}, \boldsymbol{x}')$, prior to taking observations into account. The goal is then to derive the posterior, or predictive distribution.

In practice, $\boldsymbol{g}$ is not directly observed. Following the approach of Rasmussen and Williams (2006), we use Laplace approximation to approximate the posterior distribution of $\boldsymbol{g}$ given $\mathcal{D}$. Laplace approximation is often the computationally most efficient method. It possesses attractive asymptotic properties making it accurate for large sample sizes n as those encountered in insurance studies. We refer the reader to Zilber and Katzfuss (2021) and Kündig and Sigrist (2024) for more details and alternative approaches.

Laplace approximation consists in approximating the conditional probability density function $p\left(\boldsymbol{g}\,|\mathcal{D}\right)$ of $\boldsymbol{g}$ given $\mathcal{D}$ by $q\left(\boldsymbol{g}\,|\mathcal{D}\right) \sim \mathcal{N}\left(\hat{\boldsymbol{g}}, \Sigma_{\boldsymbol{g}}\right)$, where the n-dimensional mean vector $\hat{\boldsymbol{g}}$ and the $n \times n$ covariance matrix $\Sigma_{\boldsymbol{g}}$ are such that

$$\hat{\boldsymbol{g}} = \arg\max_{\boldsymbol{g}} p\left(\boldsymbol{g}\,|\,\mathcal{D}\right) \text{ and } \Sigma_{\boldsymbol{g}}^{-1} = -\nabla\nabla \ln p\left(\boldsymbol{g}\,|\,\mathcal{D}\right)\big|_{\boldsymbol{g}=\hat{\boldsymbol{g}}}$$

with ∇ denoting the gradient operator so that $\nabla\nabla$ corresponds to the Hessian matrix.

As $p\left(\boldsymbol{g}\,|\,\mathcal{D}\right) \propto p\left(\boldsymbol{y}\,|\,\boldsymbol{g}\right) p\left(\boldsymbol{g}\,|\,\boldsymbol{X}\right)$, where "$\propto$" means "is proportional to", the mean vector $\hat{\boldsymbol{g}}$ is such that

$$\hat{\boldsymbol{g}} = \arg\max_{\boldsymbol{g}} \big(\ln p\left(\boldsymbol{y}\,|\,\boldsymbol{g}\right) + \ln p\left(\boldsymbol{g}\,|\,\boldsymbol{X}\right) \big).$$

According to model specification (2.3), $\boldsymbol{G} \sim \mathcal{N}\left(0, C(\boldsymbol{X}, \boldsymbol{X})\right)$ where $C(\boldsymbol{X}, \boldsymbol{X}) = \sigma^{*2} I_n + k(\boldsymbol{X}, \boldsymbol{X})$ is a (shrinked) Gram matrix. Hence,

$$\hat{\boldsymbol{g}} = \arg\max_{\boldsymbol{g}} \psi(\boldsymbol{g}) \text{ where } \psi(\boldsymbol{g}) \;=\; \sum_{i=1}^{n} \ln p\left(y_i\,|\,g_i\right) - \frac{1}{2}\boldsymbol{g}^{\top} C(\boldsymbol{X}, \boldsymbol{X})^{-1} \boldsymbol{g}\,. \tag{2.4}$$

2.7 Newton-Raphson algorithm

We solve the optimization problem (2.4) numerically using a Newton-Raphson algorithm. The next two results provide the gradient vector and Hessian matrix of $\psi(\cdot)$ required for implementing this method and for calculating $\Sigma_{\boldsymbol{g}}$.

2.7.1 Canonical link functions

Let us first consider GGPR models with canonical link functions. The proof of the next result is given in Appendix A.

Proposition 2.1. *For the canonical link function $l(\cdot) = \gamma'^{-1}(\cdot)$, the gradient of $\psi(\boldsymbol{g})$ is equal to*

$$\nabla\psi(\boldsymbol{g}) \;=\; \frac{\boldsymbol{\omega}}{\phi} \odot \left(\boldsymbol{y} - \gamma'\left(\boldsymbol{g}\right)\right) - C(\boldsymbol{X}, \boldsymbol{X})^{-1}\boldsymbol{g}, \tag{2.5}$$

where $\odot$ is the element-wise, or Hadamard product. The Hessian is equal to

$$\nabla\nabla\psi(\boldsymbol{g}) \;=\; -H(\boldsymbol{g}) - C(\boldsymbol{X}, \boldsymbol{X})^{-1}\,, \tag{2.6}$$

where $H(\boldsymbol{g})$ is the $n \times n$ diagonal matrix defined by

$$H(\boldsymbol{g}) = diag\left(\frac{\boldsymbol{\omega}}{\phi} \odot \gamma''\left(\boldsymbol{g}\right)\right). \tag{2.7}$$

2.7.2 Log-link function

As previously discussed, the canonical link function is mainly used in practice for the Normal, Poisson and Binomial distributions. For Gamma or Inverse Gaussian distributions, actuaries generally use a logarithmic link instead of the canonical power link function. This ensures that the conditional expectation $\mu(\boldsymbol{x})$ remains positive, whatever the combination of features. In the Binomial case, using a log-link function may result in $\mu(\boldsymbol{x})$ higher than one. Nevertheless, if $p(\boldsymbol{x})$ are small, this link may still provide the actuary with satisfactory results. An alternative consists to cap $\mu(\boldsymbol{x})$ to 1.

The next result provides the gradient and Hessian of $\psi(\cdot)$ for the logarithmic link, widely used in practice. Its proof is provided in Appendix B.

Proposition 2.2. *For the log-link function $l(\cdot) = \ln(\cdot)$, the gradient of $\psi(\boldsymbol{g})$ is equal to*

$$\nabla\psi(\boldsymbol{g}) \quad = \quad \frac{\boldsymbol{\omega}}{\phi} \odot \left(\frac{\boldsymbol{y} \odot e^{\boldsymbol{g}} - e^{2\boldsymbol{g}}}{\gamma''\left(\gamma'^{-1}\left(e^{\boldsymbol{g}}\right)\right)} \right) - C(\boldsymbol{X}, \boldsymbol{X})^{-1}\boldsymbol{g}, \tag{2.8}$$

where the division in the first term of (2.8) *is applied component-wise. The Hessian is given by* (2.6) *where $H(\boldsymbol{g})$ is the $n \times n$ diagonal matrix defined by*

$$H(\boldsymbol{g}) = diag\left(\frac{\boldsymbol{\omega}}{\phi} \odot \left(\frac{\left(\boldsymbol{y} \odot e^{2\boldsymbol{g}} - e^{3\boldsymbol{g}}\right) \odot \gamma'''\left(\gamma'^{-1}\left(e^{\boldsymbol{g}}\right)\right)}{\left(\gamma''\left(\gamma'^{-1}\left(e^{\boldsymbol{g}}\right)\right)\right)^3} - \frac{\boldsymbol{y} \odot e^{\boldsymbol{g}} - 2e^{2\boldsymbol{g}}}{\gamma''\left(\gamma'^{-1}\left(e^{\boldsymbol{g}}\right)\right)} \right) \right). \tag{2.9}$$

The gradient $\nabla\psi(\boldsymbol{g})$ and $H(\boldsymbol{g})$ in Propositions 2.1- 2.2 depend on the first-, second- and third-order derivatives of the function $\gamma(\cdot)$. These derivatives are listed in Table 2.5 for the Normal, Gamma, Inverse Gaussian, Poisson and Binomial distributions.

Y	$\gamma(z)$	$\gamma'(z)$	$\gamma''(z)$	$\gamma'''(z)$
$\mathcal{N}(\mu(\boldsymbol{x}), \frac{\sigma^2}{\omega})$	$z^2/2$	z	1	0
$\mathcal{G}\left(\omega\alpha, \omega\beta(\boldsymbol{x})\right)$	$-\ln\left(-z\right)$	$-\frac{1}{z}$	$\frac{1}{z^2}$	$-\frac{2}{z^3}$
$\mathcal{IG}\left(\mu(\boldsymbol{x}), \omega\alpha\right)$	$-\sqrt{-2z}$	$\left(-2z\right)^{-\frac{1}{2}}$	$\left(-2z\right)^{-\frac{3}{2}}$	$3\left(-2z\right)^{-\frac{5}{2}}$
$\mathcal{P}\left(\lambda(\boldsymbol{x})\,\omega\right)$	e^z	e^z	e^z	e^z
$\mathcal{B}\left(\omega, p(x)\right)$	$\ln\left(1+e^z\right)$	$\frac{e^z}{1+e^z}$	$\frac{e^z}{1+e^z} - \left(\frac{e^z}{1+e^z}\right)^2$	$\left(\frac{e^z}{1+e^z} - \left(\frac{e^z}{1+e^z}\right)^2\right)$ $\times\left(1 - 2\frac{e^z}{1+e^z}\right)$
Y	$\gamma'^{-1}(z)$	$\gamma''\left(\gamma'^{-1}\left(e^g\right)\right)$	$\gamma'''\left(\gamma'^{-1}\left(e^g\right)\right)$	
$\mathcal{N}(\mu(\boldsymbol{x}), \frac{\sigma^2}{\omega})$	z	1	0	
$\mathcal{G}\left(\omega\alpha, \omega\beta(\boldsymbol{x})\right)$	$-z^{-1}$	e^{2g}	$2e^{3g}$	
$\mathcal{IG}\left(\mu(\boldsymbol{x}), \omega\alpha\right)$	$-\frac{1}{2}z^{-2}$	e^{3g}	$3e^{5g}$	
$\mathcal{P}\left(\lambda(\boldsymbol{x})\,\omega\right)$	$\ln z$	e^g	e^g	
$\mathcal{B}\left(\omega, p(x)\right)$	$\ln\frac{z}{1-z}$	$e^g - e^{2g}$	$\left(e^g - e^{2g}\right)\left(1 - 2e^g\right)$	

Table 2.5: First-, second- and third-order derivatives of the function $\gamma(.)$ involved in $\nabla\psi(\boldsymbol{g})$ and $H(\boldsymbol{g})$.

2.8 Fitting algorithm

In applications to insurance pricing, the Poisson distribution is generally used when dealing with claim frequency while the Gamma and Inverse Gaussian distributions are preferred for claim severities. In all case, actuaries conduct the analysis with log-link function.

2.8.1 Poisson distribution with canonical log-link for claim frequencies

In the Poisson case, the gradient (2.8) and the matrix (2.7)-(2.9) take the following simple form

$$\begin{cases} \nabla\psi(\boldsymbol{g}) = \boldsymbol{\omega} \odot (\boldsymbol{y} - e^{\boldsymbol{g}}) - C(\boldsymbol{X},\boldsymbol{X})^{-1}\boldsymbol{g}\,, \\ H(\boldsymbol{g}) = \operatorname{diag}\left(\boldsymbol{\omega} \odot e^{\boldsymbol{g}}\right)\,. \end{cases}$$

2.8.2 Gamma distribution with log-link for claim severities

In the Gamma model with log-link, $\nabla\psi$ and H depend on the dispersion parameter ϕ and on the ratio of responses $\boldsymbol{y}$ on $e^{\boldsymbol{g}}$:

$$\begin{cases} \nabla\psi(\boldsymbol{g}) = \frac{\boldsymbol{\omega}}{\phi} \odot \left(\frac{\boldsymbol{y}-e^{\boldsymbol{g}}}{e^{\boldsymbol{g}}}\right) - C(\boldsymbol{X},\boldsymbol{X})^{-1}\boldsymbol{g}\,, \\ H(\boldsymbol{g}) = \operatorname{diag}\left(\frac{\boldsymbol{\omega}}{\phi} \odot \frac{\boldsymbol{y}}{e^{\boldsymbol{g}}}\right)\,. \end{cases}$$

2.8.3 Inverse Gaussian distribution with log-link for claim severities

In the Inverse Gaussian model with log-link, $\nabla\psi$ and H depend on the square of $e^{2\boldsymbol{g}}$ and on the dispersion parameter ϕ :

$$\begin{cases} \nabla\psi(\boldsymbol{g}) = \frac{\boldsymbol{\omega}}{\phi} \odot \left(\frac{\boldsymbol{y}-e^{\boldsymbol{g}}}{e^{2\boldsymbol{g}}}\right) - C(\boldsymbol{X},\boldsymbol{X})^{-1}\boldsymbol{g}\,, \\ H(\boldsymbol{g}) = \operatorname{diag}\left(\frac{\boldsymbol{\omega}}{\phi} \odot \left(\frac{2\boldsymbol{y}-e^{\boldsymbol{g}}}{e^{2\boldsymbol{g}}}\right)\right)\,. \end{cases}$$

2.8.4 Iterations

The mean vector $\hat{\boldsymbol{g}}$ in $q\left(\boldsymbol{g}\,|\mathcal{D}\right)$ is such that the gradient of $\psi(\boldsymbol{g})$ is null. It does not admit any analytical expression but a few Newton-Raphson iterations are generally enough to achieve convergence. During the j^{th}-iteration, we update the current estimate $\hat{\boldsymbol{g}}^{(j)}$ of $\hat{\boldsymbol{g}}$ as follows:

$$\begin{aligned} \hat{\boldsymbol{g}}^{(j)} &= \hat{\boldsymbol{g}}^{(j-1)} - \left(\nabla\nabla\psi\left(\hat{\boldsymbol{g}}^{(j-1)}, \hat{\phi}^{(j-1)}\right)\right)^{-1} \nabla\psi\left(\hat{\boldsymbol{g}}^{(j-1)}\right) \qquad (2.10) \\ &= \hat{\boldsymbol{g}}^{(j-1)} + \left(H\left(\hat{\boldsymbol{g}}^{(j-1)}, \hat{\phi}^{(j-1)}\right) + C(\boldsymbol{X},\boldsymbol{X})^{-1}\right)^{-1} \nabla\psi\left(\hat{\boldsymbol{g}}^{(j-1)}, \hat{\phi}^{(j-1)}\right)\,, \end{aligned}$$

where $\nabla\psi\left(\cdot\right)$ and $H(\cdot)$ are provided in Propositions 2.1- 2.2, depending upon the chosen link function.

2.8.5 Estimation of the dispersion parameter

In the Gamma and Inverse Gaussian cases, the dispersion parameter must also be estimated. The dispersion coefficient at each iteration (2.10) is generally estimated using the fitted values $\hat{y}_i^{(j)}$ as follows:

$$\hat{\phi}^{(j)} = \frac{1}{n - \text{dof}} \sum_{i=1}^{n} \omega_i \frac{\left(y_i - \hat{y}_i^{(j)}\right)^2}{v\left(\hat{y}_i^{(j)}\right)} \,,$$

where $v(\cdot)$ is the variance function in Table 2.2, dof is the number of degrees of freedom of the model (i.e. the number of hyper-parameters defining the kernel) and $\hat{y}_i^{(j)}$ is the estimated mean of the key ratio. Since obtaining $\hat{y}_i^{(j)}$ is computationally intensive (see Proposition 2.3 below for more details), we instead use an approximation based on estimated $\hat{\boldsymbol{g}}^{(j)}$:

$$\hat{\phi}^{(j)} = \frac{1}{n - \text{dof}} \sum_{i=1}^{n} \omega_i \frac{\left(y_i - l^{-1}\left(\hat{\boldsymbol{g}}^{(j)}\right)\right)^2}{v\left(l^{-1}\left(\hat{\boldsymbol{g}}^{(j)}\right)\right)} \,.$$

As shown in numerical illustrations, this approximation is sufficient to achieve excellent performances. We denote the average response by $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ and use if for initializing the Newton-Raphson algorithm with $\hat{\boldsymbol{g}}^{(0)} = l(\bar{y})$ and

$$\hat{\phi}^{(0)} = \frac{1}{n - \text{dof}} \sum_{i=1}^{n} \omega_i \frac{(y_i - \bar{y})^2}{v(\bar{y})} \,.$$

2.9 Resulting pure premium

2.9.1 New risk profile

If we see $\boldsymbol{g}$ as the realization of a random vector $\boldsymbol{G}$ and if we denote as $G' = g(\boldsymbol{x}')$ the latent component for a new risk profile $\boldsymbol{x}' \notin \boldsymbol{X}$, we have by construction that

$$\begin{pmatrix} G' \\ \boldsymbol{G} \end{pmatrix} \sim \mathcal{N}\left(\boldsymbol{0} ; \begin{pmatrix} k(\boldsymbol{x}', \boldsymbol{x}') & k(\boldsymbol{X}, \boldsymbol{x}') \\ k(\boldsymbol{X}, \boldsymbol{x}')^\top & \sigma^{*2} I_n + k(\boldsymbol{X}, \boldsymbol{X}) \end{pmatrix}\right) . \tag{2.11}$$

Using the standard properties of the multivariate Normal distribution, the conditional probability density function $p\left(g' \,|\, \mathcal{D}, \boldsymbol{g}, \boldsymbol{x}'\right)$ of G' given $\mathcal{D}$, $\boldsymbol{g}$, and $\boldsymbol{x}'$ is given by

$$p\left(g' \,|\, \mathcal{D}, \boldsymbol{g}, \boldsymbol{x}'\right) \sim \mathcal{N}\left(\mu_g\left(\boldsymbol{x}' | \mathcal{D}, \boldsymbol{g}\right) ; \sigma_g^2\left(\boldsymbol{x}' | \mathcal{D}, \boldsymbol{g}\right)\right) ,$$

where

$$\begin{aligned} \mu_g\left(\boldsymbol{x}' | \mathcal{D}, \boldsymbol{g}\right) &= \mathbb{E}\left(g(\boldsymbol{x}') | \mathcal{D}, \boldsymbol{g}, \boldsymbol{x}'\right) \\ &= k(\boldsymbol{x}', \boldsymbol{X})^\top \left(\sigma^{*2} I_n + k(\boldsymbol{X}, \boldsymbol{X})\right)^{-1} \boldsymbol{g} \end{aligned} \tag{2.12}$$

and

$$\begin{aligned} \sigma_g^2\left(\boldsymbol{x}' | \mathcal{D}, \boldsymbol{g}\right) &= \mathbb{V}\left(g(\boldsymbol{x}') | \mathcal{D}, \boldsymbol{g}, \boldsymbol{x}'\right) \\ &= k(\boldsymbol{x}', \boldsymbol{x}') - k(\boldsymbol{X}, \boldsymbol{x}')^\top \left(\sigma^{*2} I_n + k(\boldsymbol{X}, \boldsymbol{X})\right)^{-1} k(\boldsymbol{X}, \boldsymbol{x}') \,. \end{aligned} \tag{2.13}$$

2.9.2 Predictions

The Hessian of $\ln p\,(\boldsymbol{g}\,|\,\mathcal{D})$ being equal to the Hesssian of $\psi(\boldsymbol{g})$, the covariance matrix of $q\,(\boldsymbol{g}\,|\mathcal{D})$ is given by

$$\begin{aligned}\Sigma_{\boldsymbol{g}} &= \left(-\nabla\nabla \ln p\,(\boldsymbol{g}\,|\,\mathcal{D})|_{\boldsymbol{g}=\hat{\boldsymbol{g}}}\right)^{-1} \\ &= \left(-\nabla\nabla\psi(\boldsymbol{g})|_{\boldsymbol{g}=\hat{\boldsymbol{g}}}\right)^{-1} \\ &= \left(H(\hat{\boldsymbol{g}}) + C(\boldsymbol{X},\boldsymbol{X})^{-1}\right)^{-1}\,.\end{aligned} \tag{2.14}$$

Knowing the mean vector and covariance matrix of the approximation $q\,(\boldsymbol{g}\,|\mathcal{D}) \sim \mathcal{N}\,(\hat{\boldsymbol{g}}, \Sigma_{\boldsymbol{g}})$ to $p\,(\boldsymbol{g}\,|\mathcal{D})$, we can find an estimator of the pure premium as follows.

Proposition 2.3. *Let us consider an insurance policy with features $\boldsymbol{x}'$, not included in the training set. Let us define*

$$\mu_q\,(\boldsymbol{x}'|\mathcal{D}) \;=\; k(\boldsymbol{x}',\boldsymbol{X})^\top C(\boldsymbol{X},\boldsymbol{X})^{-1}\hat{\boldsymbol{g}} \tag{2.15}$$

and

$$\sigma_q^2\,(\boldsymbol{x}'|\mathcal{D}) = k(\boldsymbol{x}',\boldsymbol{x}') - k(\boldsymbol{X},\boldsymbol{x}')^\top \,\left(C(\boldsymbol{X},\boldsymbol{X}) + H(\hat{\boldsymbol{g}})^{-1}\right)^{-1} k(\boldsymbol{X},\boldsymbol{x}')\,. \tag{2.16}$$

For the Normal distribution with canonical link $l(\mu) = \mu$, the fitted value $\hat{y}'$ of the key ratio Y' is equal to

$$\hat{y}' = \mu_q\,(\boldsymbol{x}'|\mathcal{D})\;. \tag{2.17}$$

For the Gamma, Inverse Gaussian, Poisson, and Binomial distributions with log-link function $l(\mu) = \ln(\mu)$, the fitted value $\hat{y}'$ is given by

$$\hat{y}' = \exp\left(\mu_q\,(\boldsymbol{x}'|\mathcal{D}) + \frac{1}{2}\sigma_q^2\,(\boldsymbol{x}'|\mathcal{D})\right)\;. \tag{2.18}$$

In general, the fitted value $\hat{y}'$ is computed numerically by approximating the integral

$$\hat{y}' \;=\; \int l^{-1}\,(g')\;q\,(g'\,|\,\mathcal{D},\boldsymbol{x}')\,dg'\,. \tag{2.19}$$

The proof of Proposition 2.3 is given in Appendix C.

3 Selection of hyper-parameters

Despite their low number, fitting the hyper-parameters entering kernel functions may be a challenging task. Rasmussen and Williams (2006) achieve this by maximization of an approximated log-likelihood, again based on a limited Taylor expansion of $\psi(\cdot)$. This approach avoids the time-consuming calculation of the fitted values $\hat{y}$ by numerical approximation of the integral (2.19). As actuarial applications mainly rely on Gamma, Inverse Gaussian, and Poisson distributions with log-link, we can efficiently compute $\hat{y}$ from (2.18) and instead fit hyper-parameters by maximizing the deviance. Since deviance is the criterion used by actuaries for comparing and estimating models, we believe that this new approach is preferable for insurance studies.

Let us briefly recall the concept of deviance. Let $LL\left(\widehat{y}_i\right)$ be the log-likelihood of the i^{th} insurance contract key ratios, as a function of the fitted value $\widehat{y}_i$. We can get a perfect fit by setting all $\widehat{y}_i = y_i$. This configuration is called the saturated, or full model. This model is trivial and of no practical interest but since it perfectly fits data, its log-likelihood is the best one that can be achieved with the ED model under consideration. The scaled deviance D^* is defined as the likelihood ratio test statistic of the model under consideration against the saturated model:

$$D^*(y_i, \widehat{y}_i) = 2\left(LL(y_i) - LL(\widehat{y}_i)\right) .$$

As $\hat{y}_i$ is an estimate of $\mathbb{E}\left(Y_i\right)$ and $\mathbb{E}\left(Y_i\right) = \gamma'\left(\theta(\boldsymbol{x}_i)\right)$ for ED distributions, we have $\theta(\boldsymbol{x}_i) \approx \gamma'^{-1}\left(\hat{y}_i\right)$. Then, according to the definition of ED distributions, the scaled deviance is equal to

$$D^*(y_i, \widehat{y}_i) = 2\frac{\omega_i}{\phi}\left(y_i\gamma^{'-1}\left(y_i\right) - \gamma\left(\gamma^{'-1}\left(y_i\right)\right) - y_i\gamma^{'-1}\left(\widehat{y}_i\right) + \gamma\left(\gamma^{'-1}\left(\widehat{y}_i\right)\right)\right).$$

By multiplying this expression by ϕ, we get the unscaled deviance $D(y_i, \widehat{y}_i) = \phi D^*(y_i, \widehat{y}_i)$. The unscaled deviance measures the goodness of fit for estimating kernel hyper-parameters. Table 3.1 presents the unscaled deviance associated to the Normal, Gamma, Inverse Gaussian, Poisson and Binomial models. The total deviance $D(\mathcal{D})$ is simply the sum of individual unscaled deviance, that is, $D(\mathcal{D}) = \sum_{i=1}^{n} D(y_i, \widehat{y}_i)$.

	Unscaled deviance $D(y_i, \hat{y}_i)$
Normal	$\omega_i\left(y_i - \widehat{y}_i\right)^2$
Gamma	$\begin{cases} 2\omega_i\left(\frac{y_i}{\widehat{y}_i} - 1 - \ln\left(\frac{y_i}{\widehat{y}_i}\right)\right) & y_i > 0 \\ 0 & y_i = 0 \end{cases}$
Inverse Gaussian	$\omega_i\frac{(y_i - \hat{y}_i)^2}{y_i\hat{y}_i^2} \quad y_i > 0$
Poisson	$\begin{cases} 2\omega_i\left(y_i\ln y_i - y_i\ln\widehat{y}_i - y_i + \widehat{y}_i\right) & y_i > 0 \\ 2\omega_i\widehat{y}_i & y_i = 0 \end{cases}$
Binomial	$\begin{cases} 2\omega_i\left(y_i\ln\left(\frac{y_i}{\widehat{y}_i}\right) + (1 - y_i)\ln\left(\frac{1-y_i}{1-\widehat{y}_i}\right)\right) & y_i \in (0,1) \\ -2\omega_i\ln\left(1 - \widehat{y}_i\right) & y_i = 0 \\ -2\omega_i\ln\left(\widehat{y}_i\right) & y_i = 1 \end{cases}$

Table 3.1: Deviance statistics for Normal, Gamma, Inverse Gaussian, Poisson and Binomial distributions.

4 Embedding of categorical features

Until now, we have assumed that the features of an insurance contract are contained in a numeric vector $\boldsymbol{x}$ belonging to a subset $\mathcal{X} \in \mathbb{R}^m$. In this case, the Euclidean distance between two policies with respective risk profiles $\boldsymbol{x}$ and $\boldsymbol{x}'$ entering the kernels listed in Table 2.4 is well adapted to quantify their similarity and the GGPR can be used without pre-processing of data. In practice, the features of insurance policies are mostly encoded as categorical information and we must therefore convert the categorical features into relevant vectors of $\mathbb{R}^m$, to which we can apply the Euclidian distance. In this paper, we use Burt's distance as introduced in Hainaut (2019) and Jamotton et al. (2024).

Let us consider a portfolio for which features are encoded into p categorical variables which have m_k binary modalities for $k = 1, ..., p$. By binary, we mean that the modality j of the k^{th} variable is identified by an indicator variable equal to zero or one. The total number of modalities is $m = \sum_{k=1}^{p} m_k$. In the following developments, we enumerate modalities from 1 to m. The information about the portfolio can then be summarized by an $n \times m$ matrix $D = (d_{i,j})_{i=1...n,j=1...m}$. If the i^{th} policy presents the j^{th} modality then $d_{i,j} = 1$ and $d_{i,j} = 0$ otherwise. The matrix D is referred to as the disjunctive table.

Example 4.1. *Assume that risk classification in motor insurance is based on driver's gender (M=male or F=Female) and residence area (U=urban, S=suburban or C=countryside). The number of variables and modalities are respectively $p = 2$, $m_1 = 2$, $m_2 = 3$ so that $m = 5$. If the first and second policyholders are respectively a man living in a city and a woman living in the countryside, the two first lines of the matrix D are given in Table 4.1.*

	Gender		Area		
Policy	M	F	U	S	C
1	1	0	1	0	0
2	0	1	0	0	1
⋮	⋮	⋮	⋮	⋮	⋮

Table 4.1: Example of a disjunctive table for $p = 2$ categorical features with respectively $m_1 = 2$ and $m_2 = 3$ modalities.

In order to study the dependence between the modalities, we calculate the numbers $n_{i,j}$ of individuals sharing modalities i and j, for $i, j = 1, ..., m$. The $m \times m$ matrix $\boldsymbol{B} = (n_{i,j})_{i,j=1,...,m}$ is a contingency table, called the Burt matrix. The Burt matrix is computed from the disjunctive table as follows:

$$\boldsymbol{B} = \boldsymbol{D}^\top \boldsymbol{D} \,.$$

This symmetric matrix is composed of $p \times p$ blocks $\boldsymbol{B}_{k,j}$ for $k, j = 1, ..., p$. A block $\boldsymbol{B}_{k,j}$ is the contingency table that crosses the variables k and j. By construction, the sum of elements of a block $\boldsymbol{B}_{k,j}$ is equal to the total number n of policies. The sum of $n_{i,j}$ of the same row i is equal to

$$n_{i,\bullet} = \sum_{j=1,...,m} n_{i,j} = p \, n_{i,i} \,.$$

The Burt matrix being symmetric, we directly infer that

$$n_{\bullet,j} = \sum_{i=1,...,m} n_{i,j} = p \, n_{j,j} \,.$$

Blocks $\boldsymbol{B}_{k,k}$ for $k = 1, ..., p$ are diagonal, whose diagonal entries are the numbers of policies who respectively present the modalities $1, ..., m_k$, for the k^{th} variable.

Example 4.2. *Table 4.2 shows the Burt matrix for the matrix $\boldsymbol{D}$ presented in Table 4.1. Here, $n_{1,1}$ and $n_{2,2}$ respectively count the total number of men and women in the portfolio while $n_{3,3}$, $n_{4,4}$ and $n_{5,5}$ respectively count the number of policyholders living in a urban, sub-urban or rural environment. We have $n_{1,1} + n_{2,2} = n$ and $n_{3,3} + n_{4,4} + n_{5,5} = n$.*

		Gender		Area		
		M	F	U	S	C
Gender	M	$n_{1,1}$	0	$n_{1,3}$	$n_{1,4}$	$n_{1,5}$
	F	0	$n_{2,2}$	$n_{2,3}$	$n_{2,4}$	$n_{2,5}$
Area	U	$n_{3,1}$	$n_{3,2}$	$n_{3,3}$	0	0
	S	$n_{4,1}$	$n_{4,2}$	0	$n_{4,4}$	0
	C	$n_{5,1}$	$n_{5,2}$	0	0	$n_{5,5}$

Table 4.2: Burt matrix for the disjunctive Table 4.1.

We define the Chi-Square distance between rows i and i' of the Burt matrix as follows:

$$\chi^2(i, i') = \sum_{j=1}^{m} \frac{n}{n_{\bullet,j}} \left(\frac{n_{i,j}}{n_{i,\bullet}} - \frac{n_{i',j}}{n_{i',\bullet}} \right)^2, \quad i, i' \in \{1, ..., m\}.$$

Intuitively, the distance between two modalities is measured by the sum of weighted gaps between joint frequencies with respect to all modalities. Similarly, the Chi-Square distance between columns j and j' of the Burt matrix is defined by

$$\chi^2(j, j') = \sum_{i=1}^{m} \frac{n}{n_{i,\bullet}} \left(\frac{n_{i,j}}{n_{\bullet,j}} - \frac{n_{i,j'}}{n_{\bullet,j'}} \right)^2, \quad j, j' \in \{1, ..., m\}.$$

As we want to evaluate distances between policies with the Euclidean distance, the elements of the Burt matrix $n_{i,j}$ are replaced by weighted values $n_{i,j}^W$ defined as

$$n_{i,j}^W = \frac{n_{i,j}}{\sqrt{n_{i,\bullet}\, n_{\bullet,j}}}, \quad i, j = 1, ..., m\,. \tag{4.1}$$

Given that $n_{i,\bullet} = p\, n_{i,i}$ and $n_{\bullet,j} = p\, n_{j,j}$, we have that

$$n_{i,j}^W = \frac{n_{i,j}}{p\sqrt{n_{i,i}\, n_{j,j}}}, \quad i, j = 1, ..., m\,. \tag{4.2}$$

The distances between rows (i, i') and columns (j, j') of the Burt matrix become

$$\chi^2(i, i') = \sum_{j=1}^{m} \left(n_{i,j}^W - n_{i',j}^W \right)^2 \text{ and } \chi^2(j, j') = \sum_{i=1}^{m} \left(n_{i,j}^W - n_{i,j'}^W \right)^2 .$$

If $\boldsymbol{C}$ is the diagonal matrix $\boldsymbol{C} = \operatorname{diag}\left(n_{11}^{-\frac{1}{2}}, \ldots, n_{mm}^{-\frac{1}{2}}\right)$ then the weighted Burt matrix denoted as $\boldsymbol{B^W}$ is given by

$$\boldsymbol{B^W} = \frac{1}{p}\boldsymbol{C}\,\boldsymbol{B}\,\boldsymbol{C}\,.$$

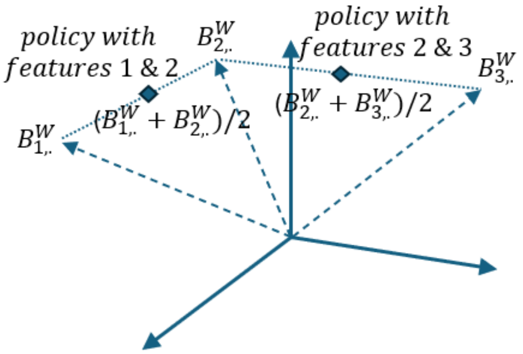
policy with features 1 & 2
$B^W_{2,.}$
policy with features 2 & 3
$B^W_{3,.}$
$(B^W_{1,.} + B^W_{2,.})/2$
$(B^W_{2,.} + B^W_{3,.})/2$
$B^W_{1,.}$

Figure 1: Illustration of the embedding with the weighted Burt matrix.

The k^{th} modality corresponds to the k^{th} row of $\boldsymbol{B^W}$, which is a vector in $\mathbb{R}^m$. The i^{th} contract with multiple modalities can then be identified by the center of gravity $\boldsymbol{D}_{i,\bullet}\boldsymbol{B^W}/p$ of points with coordinates stored in the corresponding rows of the weighted Burt matrix. This point is illustrated in Figure 1 for the case of three modalities. If each policy is defined by a subset of p =2 modalities, we represent in $\mathbb{R}^3$ as the mid point between corresponding lines of $\boldsymbol{B^W}$. The Burt's distance between the i^{th} and j^{th} policies is then the Euclidean distance between the two centers of gravity of clouds of features for each policy, that is,

$$d(\boldsymbol{x}_i, \boldsymbol{x}_j) \;\; = \;\; \left\|\boldsymbol{D}_{i,\bullet}\boldsymbol{B^W}/p - \boldsymbol{D}_{j,\bullet}\boldsymbol{B^W}/p\right\|_2 , \tag{4.3}$$

Instead of using the initial dummy vectors $(\boldsymbol{x}_i)_{i=1,\ldots,n}$, we use as entries $\boldsymbol{x}_i^{'} = \boldsymbol{D}_{i,\bullet}\boldsymbol{B^W}/p$ for the GGPR.

5 Managing large data sets

GGPR models involve a square matrix of dimension equal to the number n of records in the database. This is a serious pitfall for large data sets, such as those used in the insurance sector. It is indeed usual to work on insurance portfolios with more than one hundred thousands policies. For such a number of records, inverting the matrix $C(\boldsymbol{X}, \boldsymbol{X})$ is computationally too intensive. To address this issue, we aggregate the records into clusters and use the average of records in each cluster as input for the GGPR. Clusters are built with a batch version of the K-means algorithm. We will illustrate the efficiency of this approach in the numerical examples of the next section.

Let us consider a data set of n numeric records $X = \{\boldsymbol{x}_1, \ldots \boldsymbol{x}_n\}$ where $\boldsymbol{x}_i \in \mathbb{R}^m$ is the embedding vector of categorical features of the i^{th} contract. The corresponding responses, exposures and key ratios are respectively $n-$vectors, $\boldsymbol{r} = \{r_1, ..., r_n\}$, $\boldsymbol{\omega} = \{\omega_1, ..., \omega_n\}$ and $\boldsymbol{y} = \{y_1, ..., y_n\} = \frac{\boldsymbol{r}}{\boldsymbol{\omega}}$. The K-means algorithm searches for a partition of X into $K < n$ clusters minimizing the intra-class inertia, which measures the similarity of records inside each cluster. The average of records in a cluster is called a centroid. The centroids are $m-$dimensional vectors $\boldsymbol{c}_u = (c_1^u, \dots, c_m^u)$ for $u = 1, ..., K$. If we denote by $d(\boldsymbol{x}, \boldsymbol{x}^{'}) = ||\boldsymbol{x} - \boldsymbol{x}'||_2$ the Euclidean distance between $\boldsymbol{x}$ and $\boldsymbol{x}^{'}$, we define the clusters or classes of data S_u for $u = 1, ..., K$ as follows:

$$S_u = \{\boldsymbol{x}_i | d(\boldsymbol{x}_i, \boldsymbol{c}_u) \leq d(\boldsymbol{x}_i, \boldsymbol{c}_j) \text{ for } j = 1, ..., K\}, \quad u = 1, ..., K\,. \tag{5.1}$$

We define the barycentre of S_u as the m-dimensional vector $\boldsymbol{b}_u = \frac{1}{|S_u|} \sum_{\boldsymbol{x}_i \in S_u} \boldsymbol{x}_i$. The intra-class inertia I_a is the sum of variances inside clusters, weighted by their size, that is,

$$I_a \;=\; \frac{1}{n} \sum_{u=1}^{K} \sum_{\boldsymbol{x}_i \in S_u} d\left(\boldsymbol{x}_i, \boldsymbol{b}_u\right)^2 \,.$$

A common criterion for classification consists to look for a partition of X minimizing the intra-class inertia I_a. Finding the partition that minimizes the intra-class inertia is computationally difficult (NP-hard). The K-means is an efficient heuristic procedure using an iterative refinement technique converging quickly to a local optimum. The K-means algorithm proceeds by alternating between two steps. In the assignment step at e^{th} iteration, we associate each observation $\boldsymbol{x}_i$ with a cluster $S_u(e)$ whose centroid $\boldsymbol{c}_u(e)$ has the smallest distance $d(\boldsymbol{x}_i, \boldsymbol{c}_u(e))$. This is intuitively the nearest centroid to each observation. In the updating step, we calculate the new barycentre $\boldsymbol{b}_u(e)$ to be the centroids $\boldsymbol{c}_u(e+1)$ of observations in new clusters. The algorithm converges when the assignments no longer change. At each iteration, we can prove that the intra-class inertia is reduced. Nevertheless, we have no guarantee that the partition found in this way is a global solution. In practice, we run this algorithm several times and choose the partition with the smaller intra-class inertia. To speed up the procedure, we work with small random batches of data. The batch version of the *K-means* algorithm is provided in Appendix D.

After the partitioning of X into clusters, we assimilate the centroids to the features, $X' = \{\boldsymbol{c}_1, ..., \boldsymbol{c}_K\}$ of a reduced number of K aggregated policies. $X^{'}$ may be seen as a data set of model points, which represent locally dominant profiles of contracts. The corresponding key ratios and exposures are stored in K-vectors $\boldsymbol{y}^{'} = \{y_1^{'}, ..., y_K^{'}\}$ and $\boldsymbol{\omega}^{'} = \{\omega_1^{'}, ..., \omega_K^{'}\}$:

$$y_u^{'} = \frac{\sum_{\boldsymbol{x}_i \in S_u} r_i}{\sum_{\boldsymbol{x}_i \in S_u} \omega_i} \quad , \quad \omega_u^{'} = \sum_{\boldsymbol{x}_i \in S_u} \omega_i.$$

The reduced dataset, $(X', \boldsymbol{y}^{'}, \boldsymbol{\omega}^{'})$, serves as input for the GGPR. To conclude this section, we present in Algorithm 1 all the steps for implementing the GGPR with both embedding and clustering of a data set with categorical features. The hyper-parameters $\boldsymbol{\Theta}$ of the kernel are estimated by minimizing the unscaled deviance on the training set (e.g. with the Nelder-Mead algorithm). For large data sets, we consider a random batch of training records with a lower dimension, for computing the deviance and optimizing $\boldsymbol{\Theta}$

Algorithm 1 GGPR algorithm with embedding and clustering of data.

```
Embedding:
    Load the data set $\boldsymbol{X}$ of $n$ records with $p$ categorical features ($m$ modalities)
    Convert it into a disjunctive table $\boldsymbol{D}$ and calculate the Burt matrix $\boldsymbol{B} = \boldsymbol{D}^\top \boldsymbol{D}$
    Weighted Burt matrix $B^{\boldsymbol{W}} = \frac{1}{p}\boldsymbol{C}\,\boldsymbol{B}\,\boldsymbol{C}$ where $\boldsymbol{C} = \operatorname{diag}\left(n_{11}^{-\frac{1}{2}}..n_{mm}^{-\frac{1}{2}}\right)$
    Embedding of data: $\boldsymbol{X}' = \boldsymbol{D}B^{\boldsymbol{W}}/p$, dimension $n \times m$
Clustering:
    $K$-means applied to $\boldsymbol{X}^{'}$
    $\boldsymbol{X}''$ is the $K \times m$ matrix of model points, with key ratios $\boldsymbol{y}''$ and exposures $\boldsymbol{\omega}''$.
Main procedure:
    Select a kernel $k_{\boldsymbol{\Theta}}(.,.)$ with parameters $\boldsymbol{\Theta}$
    While $\boldsymbol{\Theta}$ is not optimal
        Compute $C(\boldsymbol{X}'', \boldsymbol{X}'') = \sigma^{*2} I_n + k(\boldsymbol{X}'', \boldsymbol{X}'')$ and $C(\boldsymbol{X}'', \boldsymbol{X}'')^{-1}$
        Estimate $\hat{\boldsymbol{g}}''$ by Newton-Raphson with $(\boldsymbol{X}'', \boldsymbol{y}'', \boldsymbol{\omega}'')$
        For all $\boldsymbol{x}' \subset \boldsymbol{X}'$ (or in a batch), compute

    $$\begin{aligned} \mu_q\left(\boldsymbol{x}'|\mathcal{D}\right) &= k(\boldsymbol{x}', \boldsymbol{X}'')^\top C(\boldsymbol{X}'', \boldsymbol{X}'')^{-1} \hat{\boldsymbol{g}}'' \\ \sigma_q^2\left(\boldsymbol{x}'|\mathcal{D}\right) &= k(\boldsymbol{x}', \boldsymbol{X}'') - k(\boldsymbol{X}'', \boldsymbol{x}')^\top \left(C(\boldsymbol{X}'', \boldsymbol{X}'') + H(\hat{\boldsymbol{g}}'')^{-1}\right)^{-1} k(\boldsymbol{X}'', \boldsymbol{x}') \end{aligned}$$

        and estimate the key ratio

            $$\hat{y}' = \exp\left(\mu_q\left(\boldsymbol{x}'|\mathcal{D}\right) + \frac{1}{2}\sigma_q^2\left(\boldsymbol{x}'|\mathcal{D}\right)\right)$$

        Compute the unscaled deviance $D(y, \hat{y}')$ and update $\boldsymbol{\Theta}$.
    End while
```

6 Numerical illustrations

This section applies Algorithm 1 on two publicly available data sets with different sizes. The first one concerns motor third-party liability (MTPL) insurance claims. It corresponds to a large portfolio of an insurance company operating in France. It is included in the `R` library `CASdatasets` contributed by Dutang and Charpentier (2020). The second data set focuses on motorcycle insurance claim (MCC) from the Swedish company WASA. The number of policies is smaller compared to French MTPL data.

The implementation is done in `Python` and we use the library `cuda` to parallelize matrix product operations and kernel evaluations. The code is run on a laptop with an Intel Core Ultra 7 processor, 32 GB of RAM, and an Nvidia RTX 2000 GPU. In this section, we band all continuous features to make them categorical. This allows us to work with homogeneous data types to demonstrate the efficiency of the embedding method proposed in Section 4. GGPR analysis combining numerical and categorical features is discussed in the final section.

6.1 MTPL database

6.1.1 Available features

This data set is extensively described in Chapter 13 of Wüthrich and Merz (2023). It provides information about claim frequencies and severities for 678 013 MTPL insurance contracts, allotted in 5 similar folds of 135 602 policies. We categorize all numerical features and Table 6.1 reports their respective levels after categorization as well as the corresponding numbers of modalities.

Features	Modalities	Number
Area	A to F	6
Vehicle power	[0,20), [20,40),..., [80,100]	8
Vehicle age	[0,5), [5,10),...., [20,100]	4
Driver age	[18,28), [28,38),..., [88,100]	8
Bonus Malus	[50,60), [60,70),...,[120,230]	8
Vehicle Brand	B1, B2, ...	11
Vehicule Gas	Diesel or Regular	2
Density	[0,20), [20, 40),... ,[80,100]	5
Region	R11, R21,...	22

Table 6.1: Categorized features and associated levels for MTPL insurance data.

6.1.2 Shrinkage and clustering

The shrinkage parameter σ^* is set to 0.01 but it could have been set to zero in this particular example as we do not observe numerical instabilities. It is nevertheless useful to assign to σ^* a small but strictly positive value to ensure that the covariance matrix remains well conditioned at each step of the estimation algorithm. If not, this matrix could become ill conditioned in the iterations which slows down or even prevents convergence. Specifying a positive shrinkage parameter σ^* makes GGPR model calibration more robust.

We run 10 times the K-means algorithm on the whole data set (no batch) and select the partition with the lowest intra-class inertia.

6.1.3 Claim severities

Let us first consider claim severities. The data set contains 24 944 records. We respectively consider folds 1 to 4 (19 942 claims) as training set and fold 5 (5 002 claims) as validation set. After converting into dummy features, we have 75 binary explanatory variables. The GLM is compared to the GGPR with RBF, RQ, M32 and M52 kernels listed in Table 2.4.

Table 6.2 reports the deviances, log-likelihoods and AIC of the Gamma GGPR model with log-link considering various kernels and numbers of clusters. The analysis of deviances and log-likelihoods on the training set reveals that the GGPR achieves a better goodness of fit than the GLM. Regardless of the kernel, increasing the number of clusters reduces the deviance. The lowest one being attained with a RQ kernel. In terms of AIC, the performance of the GGPR is remarkable given its low number of parameters (2 or 3). The validation set reveals that GGPR slightly overfits the data since the deviance is higher than with the GLM. Considering fewer than 600 clusters decreases the deviance on the validation set but raises it on the training set. For instance, with the RQ kernel and only 150 clusters, the training deviance moves up to 30 631 while the validation deviance falls to 9 024.

Gamma		Training set			Validation set			
	K	$D(\mathcal{D})$	log-like.	AIC	$D(\mathcal{D})$	log-like.	AIC	Time
RBF	600	31685.09	-196950.53	393905.07	8708.36	-52818.0	**105640.0**	92.62
	800	31576.57	-197387.3	394778.59	9512.13	-55091.54	110187.08	173.26
	1000	31302.01	-200318.94	400641.88	10144.66	-58292.22	116588.45	262.27
RQ	600	29272.1	-196186.16	392378.32	10224.85	-59542.11	119090.21	69.82
	800	28191.45	-192446.46	384898.92	10665.41	-61569.56	123145.12	125.65
	1000	27209.87	-188490.95	376987.89	10364.51	-60633.52	121273.05	199.02
M32	600	29598.39	-196966.1	393936.2	10132.6	-58904.17	117812.34	70.64
	800	28615.14	-190815.71	381635.42	10934.76	-62057.93	124119.86	98.21
	1000	27616.37	-188580.93	377165.87	10663.19	-61342.81	122689.63	144.28
M52	600	29979.38	-194723.58	389451.17	10777.34	-60628.78	121261.56	60.67
	800	28958.18	-188593.13	377190.26	11444.76	-63002.58	126009.15	103.91
	1000	27911.23	-187527.1	**375058.19**	10936.57	-61646.83	123297.67	146.99
GLM	n.a.	31689.53	-200943.82	402019.64	8456.44	-52576.03	105284.06	n.a.

Table 6.2: Gamma model with log-link function for MTPL claim severities. Comparison of deviances, log-likelihoods, AIC of GGPR for various numbers of clusters (K), and GLM. The column "Time" reports the duration in seconds for optimizing kernel parameters.

Table 6.3 reports the deviances, log-likelihoods and AIC of the Inverse Gaussian GGPR model with log-link with different kernel functions and various numbers of clusters. Based on log-likelihoods, the Inverse Gaussian model better fits data than the Gamma model. The kernel significantly influences the goodness of fit. On the training data set, the M32 kernel leads to the lowest deviance and highest log-likelihood. Nevertheless, on the validation set, the best fit is achieved with the RBF and 600 clusters. As it was the case with the Gamma model, we observe some overfitting that can be mitigated by considering fewer clusters. For the RQ kernel and K =400, the validation deviance falls to 10.56 whereas the training deviance slightly climbs to 47.58.

Inv. Gaus.		Training set			Validation set			
	K	$D(\mathcal{D})$	log-like.	AIC	$D(\mathcal{D})$	log-like.	AIC	Time
RBF	600	47.11	-177631.47	355266.93	10.58	-45399.06	**90802.11**	83.1216
	800	47.11	-179287.8	358579.6	10.87	-46766.85	93537.7	127.6252
	1000	47.1	-182014.04	364032.08	11.09	-48183.1	96370.19	145.4711
RQ	600	45.86	-175511.44	351028.88	11.08	-48799.85	97605.7	94.1315
	800	45.63	-174582.09	349170.18	11.64	-51121.66	102249.32	157.4251
	1000	45.27	-173497.29	347000.58	11.51	-50752.94	101511.89	230.4646
M32	600	46.2	-174116.17	348236.34	11.4	-49753.33	99510.66	61.7404
	800	45.96	-173135.31	346274.63	11.98	-51813.65	103631.3	101.1904
	1000	45.57	-173076.72	**346157.44**	11.76	-51338.31	102680.62	140.5164
M52	600	46.56	-173550.09	347104.17	11.68	-50690.08	101384.17	73.8574
	800	46.32	-172835.34	345674.68	12.23	-52233.67	104471.33	129.127
	1000	45.92	-173004.54	346013.08	11.83	-51259.79	102523.57	159.288
GLM	n.a.	46.96	-178563.55	357259.09	10.59	-45703.68	91539.35	n.a.

Table 6.3: Inverse Gaussian model with log-link for MTPL claim severities. Comparison of deviances, log-likelihoods, AIC of GGPR for various cluster sizes (K), and GLM. The column "Time" reports the duration in seconds for optimizing kernel parameters.

Figure 2 compares the histograms of estimated mean claim severities (on the log-scale) computed from GLM and GGPR (with $K = 600$, RQ kernel and Inverse Gaussian distribution), on the training and validation sets. Table 6.4 compares the moments and percentiles of fitted severities. It reveals that the GGPR yields more dispersed predictions than the GLM for the training set. We also observe that GGPR underestimates the average severity compared to the GLM which itself falls below the observed average severity. In the latter case, this is because the log-link is not the canonical link for the Inverse Gaussian distribution so that global balance does not necessarily hold despite the inclusion of an intercept in the GLM score. Figure 3 shows QQ-plots of predicted claim severities by GGPR and GLM. On the training set, GGPR quantiles corresponding to high probabilities are clearly greater than those of the GLM. This trend is less pronounced on the validation set.

	Method	Mean	Standard	Percentiles	
			deviation	5%	95%
Training	GGPR	2016	1081	1190	3419
	GLM	2106	843	1328	3850
	Data	2233	30857	80	4632
Validation	GGPR	1991	854	1201	3350
	GLM	2096	872	1331	3855
	Data	2180	19886	80	4718

Table 6.4: Mean, standard deviation, 5% and 95% percentiles of expected claim amounts.

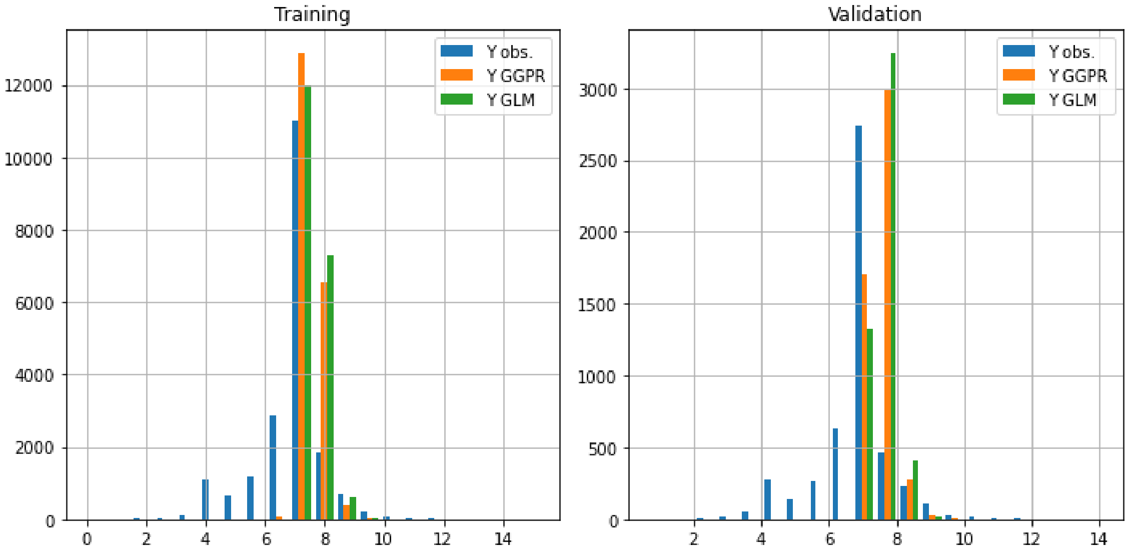

Figure 2: Histogram (20 bins) of observed and predicted claims severities on the training and validation sets. GGPR predictions are computed with $K = 600$ clusters and the RQ kernel.

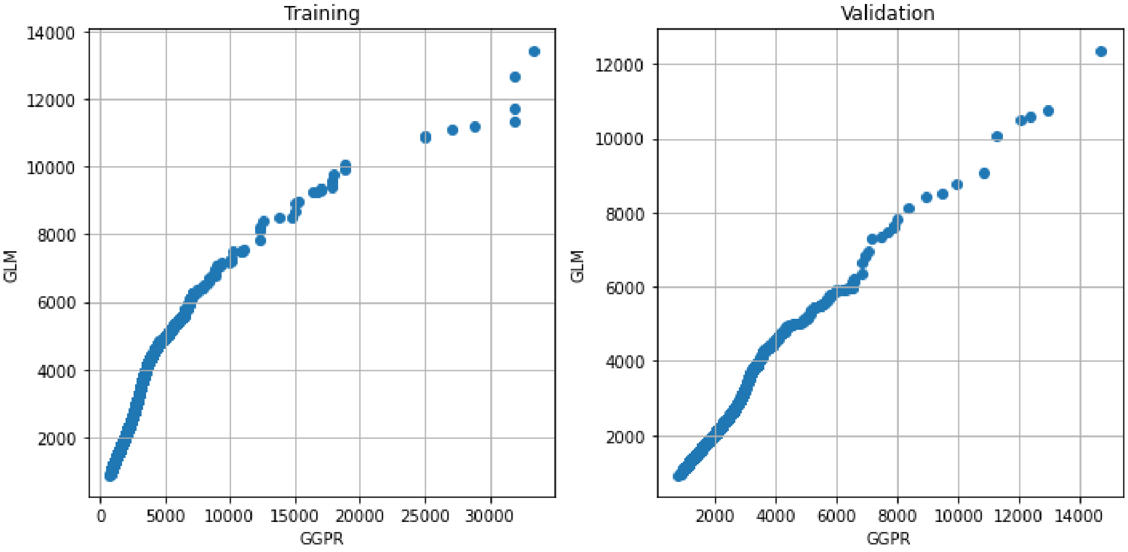

Figure 3: QQ-plots of GGPR and GLM predictions with $K = 600$ clusters and the RQ kernel.

6.1.4 Claim frequencies

Next, we fit a Poisson model with log-link to claim frequencies. We respectively consider the first and second folds (with 135 602 policies in each fold) as the training and validation sets. We run the K-means algorithm 10 times with batches of 10 000 contracts and select the partition with the lowest intra-class inertia. The hyper-parameters are estimated by minimizing the deviance for a random sample of 10 000 policies from the training set. Table 6.5 reports the deviances, log-likelihoods, and AIC of the Poisson GGPR model. Given the large number of policies, we allow for more clusters compared to the modeling of severities. Again, the GGPR outperforms the GLM, even if the gain in log-likelihood or deviance is less impressive than the one obtained with claim severities. On the training set, the GGPR slightly underperforms, possibly due to a small overfit on the validation set. However, we need to recall that GGPR is mainly a non-parametric method with only two or three degrees of freedom. From this point of view, its capacity for modeling is impressive compared to a fully parametric method such as the GLM.

Poisson		Training set			Validation set			
	K	$D(\mathcal{D})$	log-like.	AIC	$D(\mathcal{D})$	log-like.	AIC	Time
RBF	3000	32555.46	-21304.78	42613.56	32692.44	-21422.19	**42848.38**	711.36
	4000	32550.19	-21302.15	42608.3	32803.15	-21477.54	42959.09	942.58
	5000	32401.7	-21227.9	42459.8	32766.48	-21459.21	42922.42	2288.75
RQ	3000	32506.87	-21280.48	42566.97	32792.73	-21472.33	42950.67	505.46
	4000	32182.74	-21118.42	42242.85	32863.72	-21507.83	43021.67	890.25
	5000	31812.22	-20933.16	41872.32	32999.2	-21575.57	43157.14	1471.03
M32	3000	32288.88	-21171.49	42346.98	32859.32	-21505.63	43015.26	637.64
	4000	32180.93	-21117.51	42239.03	33043.8	-21597.87	43199.74	1420.29
	5000	31948.72	-21001.41	**42006.82**	32969.02	-21560.48	43124.96	2178.37
M52	3000	32371.29	-21212.7	42429.39	32858.09	-21505.02	43014.03	538.43
	4000	32281.31	-21167.71	42339.42	33041.4	-21596.67	43197.35	1109.08
	5000	32067.1	-21060.6	42125.2	32978.14	-21565.04	43134.08	1727.06
GLM	n.a.	32520.5	-21287.3	42706.6	32503.99	-21327.96	42787.93	n.a.

Table 6.5: Poisson model with log-link for MTPL claim frequencies. Comparison of deviances, log-likelihoods, AIC of GGPR for various cluster sizes (K), and GLM. The column "Time" reports the duration for optimizing kernel parameters.

Table 6.6 reports the means, standard deviations and 5%-95% percentiles of expected claim frequencies computed with the GGPR and GLM (RBF kernel and K =3000). On both training and validation sets, the Poisson GGPR yields predictions with slightly higher mean and standard deviation than the GLM. Both models fail to capture the high volatility of claim numbers. The QQ-plots in Figure 4 reveal that quantiles of GGPR and GLM predictions are similar for claim frequencies less than 0.60. Notice that the vast majority of contracts are below this threshold.

	Method	Mean	Standard	Percentiles	
			deviation	5%	95%
	GGPR	0.0887	0.0569	0.0340	0.2002
Training	GLM	0.0780	0.0524	0.0335	0.1737
	Data	0.1211	2.2772	0.0000	0.0000
	GGPR	0.0888	0.0568	0.0341	0.2001
Validation	GLM	0.0781	0.0525	0.0336	0.1732
	Data	0.1161	1.7955	0.0000	0.0000

Table 6.6: Mean, standard deviation, 5% and 95% percentiles of expected claim frequencies.

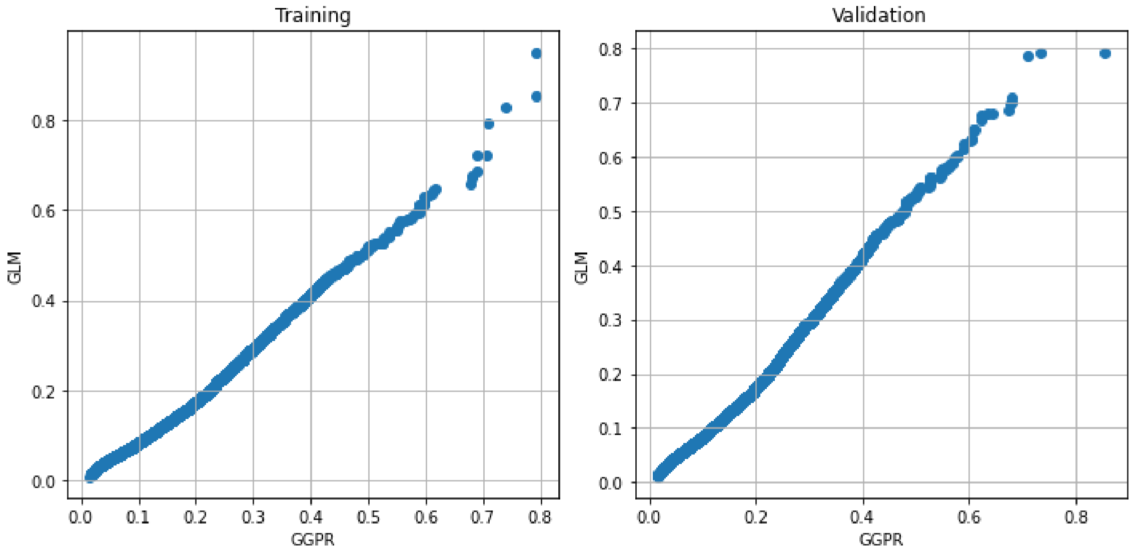

Figure 4: QQ plots of GGPR and GLM predictions with $K = 3000$ clusters and the RBF kernel.

6.2 MCC data set

6.2.1 Available features

This data set is available on the companion website of the book by Ohlsson and Johansson (2010). We refer the reader to Chapter 2 of this book for a thorough description. Claim frequencies and severities are recorded for 62 436 motorcycle insurance contracts over the period 1994-1998. As explained before, we categorize all numerical variables. Table 6.7 lists the features after categorization and the number of modalities per feature.

Features	Modalities	Number
Gender	M , F	2
Area	1 to 7	7
Vehicle power	1 to 7	7
Bonus class	1 to 7	7
Driver age	[16,26), [26,36),...., [66,100]	6
Vehicle age	[0,5), [5,10), [10,20), [20,100]	4

Table 6.7: Categorized features used as explanatory variables for MCC insurance data.

6.2.2 Shrinkage and clustering

The shrinkage parameter σ^* is set to 0.01 but we do not observe numerical instabilities with lower values. We run 10 times the K-means algorithm on the whole data set (no batch) and select the partition with the lowest intra-class inertia.

6.2.3 Claim severities

First, we consider claim severities. The data set contains 670 claims. The learning and validation sets respectively contain 536 and 134 claims. We have 33 binary explanatory variables. Table 6.8 reports goodness-of-fit statistics for the Gamma model with log-link. On the training set, the GGPR achieves lower deviance with at least 100 clusters. As with the MTPL data set, increasing the number of clusters reduces the deviance. From an AIC perspective, the GGPR is more efficient than the GLM regardless of the tested model, since it has two or three hyper-parameters. On the validation set, the GGPR deviance is higher than that of the GLM, indicating a tendency to overfit the training data. The model with the lowest AIC is the RBF GGPR with only 150 clusters.

Table 6.9 reports goodness-of-fit statistics for the Inverse Gaussian model with log-link. The log-likelihoods are lower compared to the Gamma model. On the training set, the GGPR again achieves a lower deviance with a sufficient number of clusters. Nevertheless, the Nelder-Mead algorithm converges to a local minimum during the fit of hyper-parameters of M32 and M52 kernels when K=150. On the training set, the deviances of GGPR models with 100 clusters are close to the GLM deviance. On this set, the lowest AIC value is attained with RQ kernel and $K = 100$.

Figure 5 compares the histograms of predicted mean claim severities computed with GLM and GGPR ($K = 50$, RBF kernel and Gamma distribution), on the training and validation sets. Table 6.10 compares the moments and percentiles of expected claim amounts. The standard deviations and 5%-95% percentiles of GGPR predicted responses are a bit lower than those of the GLM. The QQ-plots displayed in Figure 6 of GGPR and GLM expected claim amounts confirm this trend.

Gamma		Training set			Validation set			
	K	$D(\mathcal{D})$	log-like.	AIC	$D(\mathcal{D})$	log-like.	AIC	Time
RBF	50	873.6	-5811.83	11627.65	228.92	-1468.79	**2941.59**	1.33
	100	772.4	-5759.47	11522.94	265.87	-1501.41	3006.82	1.28
	150	742.15	-5747.65	11499.3	254.23	-1494.93	2993.85	2.83
RQ	50	845.12	-5790.62	11587.25	229.42	-1469.05	2944.11	3.14
	100	721.31	-5737.99	11481.98	247.29	-1486.74	2979.48	2.41
	150	687.75	-5723.84	**11453.67**	351.01	-1711.03	3428.06	3.10
M32	50	852.86	-5797.58	11599.16	232.63	-1470.28	2944.56	1.40
	100	744.37	-5749.06	11502.12	271.9	-1506.05	3016.1	1.01
	150	707.89	-5736.89	11477.78	371.21	-1715.59	3435.17	1.16
M52	50	875.11	-5811.21	11626.41	244.56	-1475.61	2955.22	1.35
	100	779.15	-5770.3	11544.59	292.49	-1524.36	3052.73	0.85
	150	729.82	-5754.08	11512.17	390.15	-1721.15	3446.29	1.04
GLM	n.a.	838.76	-5788.27	11632.53	219.44	-1465.74	2987.49	n.a.

Table 6.8: Gamma model with log-link for MCC claim severities. Comparison of deviances, log-likelihoods, AIC of GGPR for various cluster sizes (K), and GLM. The column "Time" reports the duration for optimizing kernel parameters.

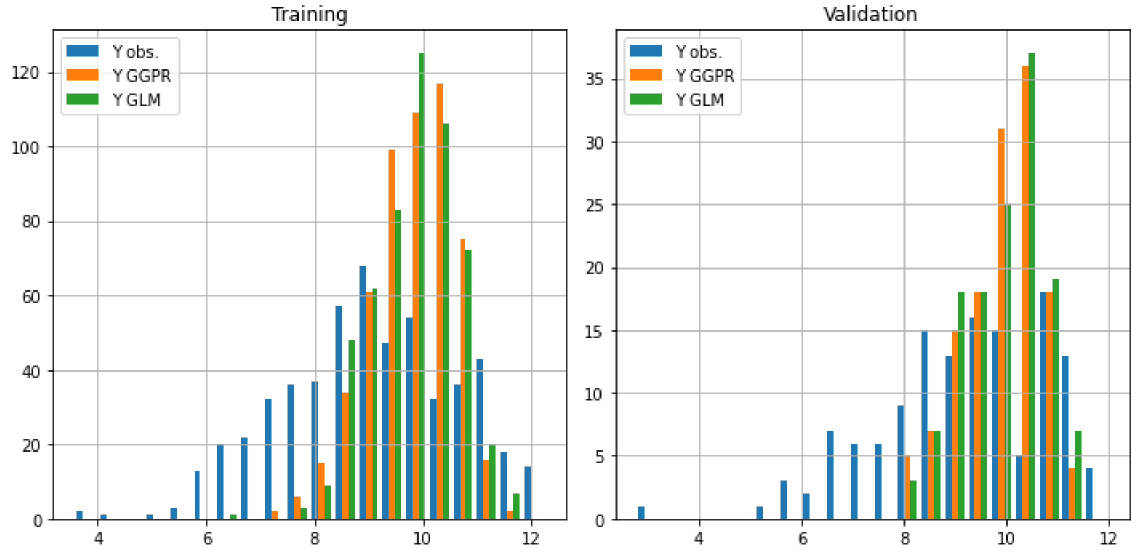

Figure 5: Histogram (20 bins) of observed and predicted log-claims on the training and validation sets. GGPR predictions are computed with $K = 100$ clusters and the RQ kernel.

Inv. Gaus.		Training set			Validation set			
	K	$D(\mathcal{D})$	log-like.	AIC	$D(\mathcal{D})$	log-like.	AIC	Time
RBF	50	0.23	-6488.66	12981.32	0.10	-2043.71	4091.42	1.76
	100	0.21	-6138.7	12281.4	0.12	-1559.85	3123.71	1.21
	150	0.21	-6175.35	12354.71	0.10	-1845.32	3694.63	1.97
RQ	50	0.22	-5980.62	**11967.24**	0.11	-1591.66	3189.33	2.22
	100	0.21	-6027.57	12061.15	0.12	-1564.01	3134.02	2.34
	150	0.19	-6025.54	12057.08	0.11	-1554.69	**3115.38**	2.93
M32	50	0.21	-6317.18	12638.35	0.10	-1879.68	3763.36	1.9928
	100	0.20	-6104.74	12213.49	0.11	-1562.43	3128.86	2.1113
	150	0.21	-6257.09	12518.17	0.10	-1771.27	3546.55	2.3633
M52	50	0.21	-6162.0	12328.01	0.1	-1742.72	3489.45	3.1729
	100	0.21	-6109.35	12222.71	0.12	-1559.02	3122.03	1.4672
	150	0.21	-6366.4	12736.8	0.1	-1830.76	3665.51	1.7457
GLM	n.a.	0.20	-6251.03	12558.06	0.11	-1557.36	3170.72	n.a.

Table 6.9: Inverse Gaussian model with log-link for MCC claim severities. Comparison of deviances, log-likelihoods, AIC of GGPR for various cluster sizes (K), and GLM. The column "Time" reports the duration for optimizing kernel parameters.

	Method	Mean	Standard	Percentiles	
			deviation	5%	95%
	GGPR	22988	16022	4725	53023
Training	GLM	24195	18515	4927	57164
	Data	23585	34528	576	90987
	GGPR	24693	15647	4761	52611
Validation	GLM	25827	18043	4958	62341
	Data	24624	30369	628	79021

Table 6.10: Mean, standard deviation, 5% and 95% percentiles of expected claim severities.

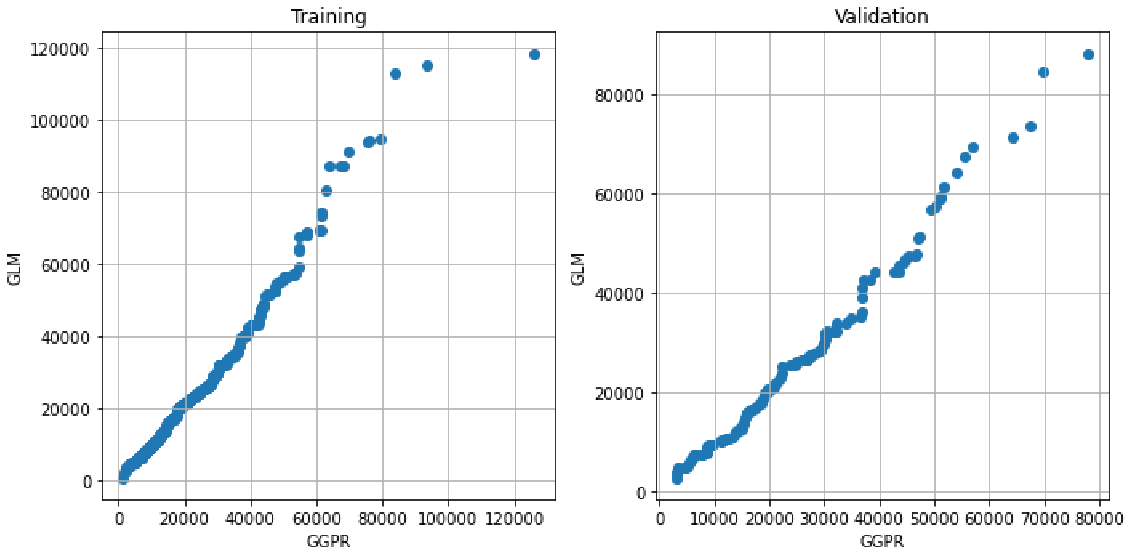

Figure 6: QQ plots of GGPR and GLM predictions with $K = 100$ clusters and the RQ kernel.

6.2.4 Claim frequencies

We next estimate a Poisson model with log-link for claim frequencies. We randomly split the data set into training (80%) and validation (20%) sets. The hyper-parameters are estimated by minimizing the deviance for a random sample of 10 000 policies from the training set. Table 6.11 reports goodness-of-fit statistics. The GGPR outperforms the GLM in most configurations both on training and validation sets. The M32 kernel with 500 clusters offers a good trade-off from this perspective.

Poisson		Training set			Validation set			
	K	$D(\mathcal{D})$	log-like.	AIC	$D(\mathcal{D})$	log-like.	AIC	Time
RBF	250	4680.9	-2877.81	5759.63	1185.84	-729.84	1463.68	9.215
	500	4667.23	-2870.98	5745.96	1170.97	-722.41	1448.81	25.089
	750	4619.63	-2847.18	5698.36	1183.48	-728.66	1461.32	78.4479
RQ	250	4580.51	-2827.62	5661.24	1170.59	-722.21	1450.43	26.4901
	500	4568.92	-2821.83	5649.65	1171.36	-722.6	1451.2	70.6792
	750	4508.95	-2791.84	**5589.68**	1187.01	-730.43	1466.85	185.7749
M32	250	4611.11	-2842.92	5689.84	1176.02	-724.93	1453.86	12.6059
	500	4607.43	-2841.08	5686.16	1168.46	-721.15	**1446.3**	31.2037
	750	4554.12	-2814.42	5632.85	1186.58	-730.21	1464.42	122.8753
M52	250	4635.59	-2855.16	5714.32	1180.92	-727.38	1458.76	6.6902
	500	4628.73	-2851.73	5707.46	1168.64	-721.24	1446.48	26.3908
	750	4581.17	-2827.95	5659.9	1185.62	-729.73	1463.47	74.99
GLM	n.a.	4607.52	-2841.12	5738.25	1198.63	-736.24	1528.47	n.a.

Table 6.11: Poisson model with log-link for MCC claim frequencies. Comparison of deviances, log-likelihoods, AIC of GGPR for various cluster sizes (K), and GLM. The column "Time" reports the duration for optimizing kernel parameters.

Table 6.12 presents the moments and percentiles of predicted frequencies with GGPR (M32 kernel with 500 clusters) and GLM. Figure 7 displays QQ-plots of predicted expected claim frequencies. These plots emphasize that the GGPR model yields higher quantiles than the GLM.

	Method	Mean	Standard	Percentiles	
			deviation	5%	95%
Training	GGPR	0.0171	0.0239	0.0020	0.0607
	GLM	0.0133	0.0200	0.0014	0.0488
	Data	0.0254	0.6286	0.0000	0.0000
Validation	GGPR	0.0173	0.0237	0.0021	0.0619
	GLM	0.0133	0.0198	0.0014	0.0488
	Data	0.0397	1.7004	0.0000	0.0000

Table 6.12: Mean, standard deviation, 5% and 95% percentiles of expected frequencies.

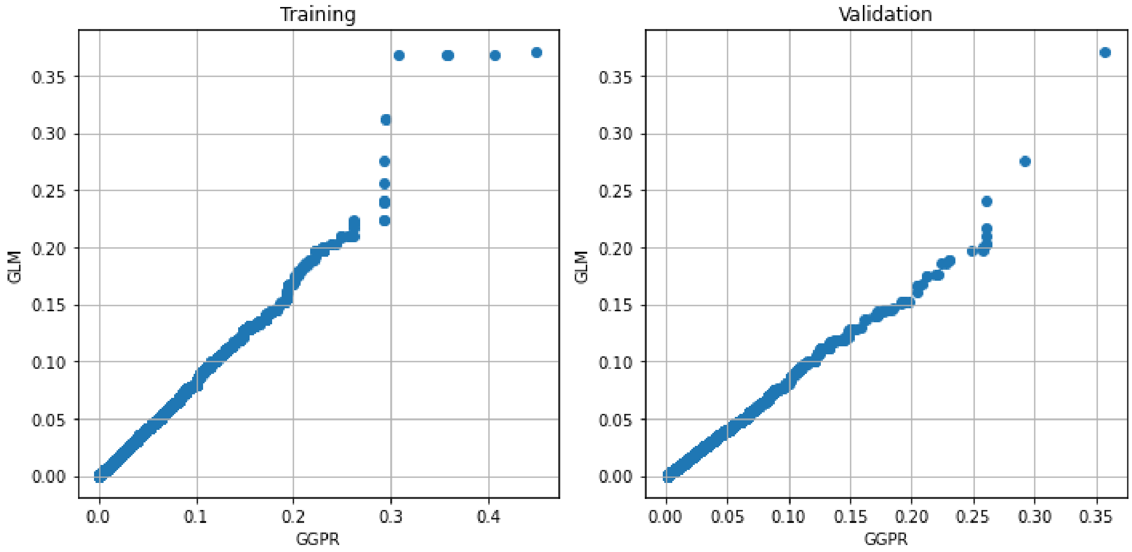

Figure 7: QQ plots of GGPR and GLM predictions with $K = 500$ clusters and the M32 kernel.

7 Discussion

This paper introduced actuaries to GGPR model for risk classification. It innovates by adapting standard GGPR approach developed after Chan and Dong (2011) to reflect insurance data specific traits. Precisely, (i) risk exposures are accounted for, (ii) hyper-parameters entering kernel functions are estimated by minimizing the deviance, (iii) categorical features are included in the analysis by using Burt's distance to assess proximity, and (iv) K-means clustering converts the initial data set into a limited number of "model points" to reduce the dimension of the problem.

The numerical illustrations performed on two publicly available insurance data sets demonstrate the excellent performances of GGPR compared to GLMs, with the advantage that GGPR is in essence non-parametric and does not require to rigidly structure the score beforehand. GGPR automatically accounts for possible interactions present in the data and can be used as an agnostic preliminary risk evaluation.

Notice also that GGPR provides the actuary with a full predictive distribution, not only point estimates. This helps the analyst to evaluate the confidence in the resulting pure premiums. Interpretation tools developed for machine learning models (like PDP, ICE or feature importance implemented in `scikit learn`, for instance) apply to GGPR outputs. This is also the case for interaction plots.

The embedding technique based on the scaled Burt matrix used for categorical features goes back to multiple correspondence analysis proposed by the French statistical school in the 1960s. See e.g. Hjellbrekke (2018) for a general account. It turns out to be very effective to assess the association between categorical variables and to deal with categorical information in insurance studies, as demonstrated by Hainaut (2019). There are of course alternatives, like the neural embedding networks mentioned in the introductory section. Another approach has been recently applied by Fernandes Machado et al. (2025) in the context of optimal transport. These authors transform categorical variables into continuous ones using their compositional representation. An assessment of the respective merits of the different embedding techniques for categorical information, depending on the context, may be relevant to guide practitioners.

In this paper, all continuous features have been banded to make them categorical and assess the performances of the proposed embedding technique. It is nevertheless possible to perform a GGPR analysis including both types of features, continuous ones and categorical ones. The kernel in the GGPR model is then the product of one kernel for numerical features and another kernel for categorical features. We leave further investigation of this approach for a subsequent work.

Acknowledgement

The authors thank the FNRS (Fonds de la recherche scientifique) for financial support through the EOS project Asterisk (research grant 40007517).

Disclosure statement

No potential conflict of interest was reported by authors.

References

[1] Avanzi, B., Taylor, G., Wang, M., Wong, B. (2024). Machine learning with high-cardinality categorical features in actuarial applications. ASTIN Bulletin 54, 213-238.

[2] Boskov, M., Verrall, R.J. (1994). Premium rating by geographic area using spatial models. ASTIN Bulletin 24, 131-143.

[3] Carlin, L., Benjamini, Y. (2025). CardiCat: A variational autoencoder for high-cardinality tabular data. arXiv preprint arXiv:2501.17324.

[4] Chan, A.B., Dong D. (2011). Generalized Gaussian process models. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2011, pp. 2681-2688.

[5] Denuit, M., Lang, S. (2004). Nonlife ratemaking with Bayesian GAMs. Insurance: Mathematics and Economics 35, 627-647.

[6] Denuit, M., Hainaut, D., Trufin, J. (2019). Effective Statistical Learning Methods for Actuaries Volume 1: GLM and Extensions Springer Actuarial Lecture Notes Series.

[7] Dimakos, X.K., Rattalma, A.F. (2002). Bayesian premium rating with latent structure. Scandinavian Actuarial Journal 2002, 162-184.

[8] Dutang, C., Charpentier, A. (2020). CASdatasets: Insurance datasets. R package version 1.0-11.

[9] Fernandes Machado, A. F., Charpentier, A., Gallic, E. (2025). Optimal transport on categorical data for counterfactuals using compositional data and Dirichlet transport. arXiv preprint arXiv:2501.15549.

[10] Hainaut, D. (2019). Self Organizing Maps for non-life insurance. European Actuarial Journal 9, 173-207.

[11] Hjellbrekke, J. (2018). Multiple Correspondence Analysis for the Social Sciences. Routledge.

[12] Jamotton, C., Hainaut, D., Hames, T. (2024). Insurance analytics with clustering techniques. Risks 12, 141.

[13] Klein, N., Denuit, M., Lang, S., Kneib, Th. (2014). Nonlife ratemaking and risk management with Bayesian additive models for location, scale and shape. Insurance: Mathematics and Economics 55, 225-249.

[14] Kündig, P., Sigrist, F. (2024). Iterative methods for Vecchia-Laplace approximations for latent Gaussian process models. Journal of the American Statistical Association, in press.

[15] Makov, U.E. (2002). Principal applications of Bayesian methods in actuarial science: A perspective. North American Actuarial Journal 5, 53-57.

[16] Ohlsson, E., Johansson, B. (2010). Non-Life Insurance Pricing with Generalized Linear Models. Springer.

[17] Rasmussen, C.E., Williams, C.K.I. (2006). Gaussian Processes for Machine Learning. MIT Press.

[18] Shi, P., Shi, K. (2023). Non-life insurance risk classification using categorical embedding. North American Actuarial Journal 27, 579-601.

[19] Wang, R., Shi, H., Cao, J. (2025). A nested GLM framework with neural network encoding and spatially constrained clustering in non-life insurance ratemaking. North American Actuarial Journal, in press.

[20] Wüthrich, M., Merz, M. (2023). Statistical Foundations of Actuarial Learning and its Applications. Springer.

[21] Zilber, D., Katzfuss, M. (2021). Vecchia–Laplace approximations of generalized Gaussian processes for big non-Gaussian spatial data. Computational Statistics & Data Analysis 153, 107081.

A Proof of Proposition 2.1

Since $\mu(\boldsymbol{x}_i) = l^{-1}(g_i)$ and $\theta_i = \gamma'^{-1}(l^{-1}(g_i))$, the conditional probability density function of y_i given g_i is retrieved from (2.1):

$$p(y_i \mid g_i) = \exp\left\{\frac{y_i\,\gamma'^{-1}(l^{-1}(g_i)) - \gamma(\gamma'^{-1}(l^{-1}(g_i)))}{\phi/\omega} + c(y, \phi, \omega)\right\}. \tag{A.1}$$

By definition, the canonical link is such that $l(\cdot) = \gamma'^{-1}(\cdot)$ and the conditional probability density function of y_i given g_i becomes

$$p(y_i \mid g_i) = \exp\left\{\frac{y_i\,g_i - \gamma(g_i)}{\phi/\omega} + c(y, \phi, \omega)\right\}. \tag{A.2}$$

The first- and second-order derivatives of the log of (A.2) with respect to g_i are respectively equal to

$$\frac{d}{dg_i}\ln p(y_i \mid g_i) = \frac{\omega}{\phi}(y_i - \gamma'(g_i)) \text{ and } \frac{d^2}{dg_i^2}\ln p(y_i \mid g_i) = -\frac{\omega}{\phi}\gamma''(g_i)\,.$$

The expression (2.5) of the gradient follows from (2.4). The matrix $H(\boldsymbol{g})$ in Equation (2.7) is the diagonal matrix of $-\frac{d^2}{dg_i^2}\ln p(y_i \mid g_i)$. If we derive twice (2.4), we retrieve (2.7). This ends the proof.

B Proof of Proposition 2.2

As $l^{-1}(g) = e^g$, (A.1) shows that

$$p(y_i \mid g_i) \propto \exp\left\{\frac{y_i\,\gamma'^{-1}(e^{g_i}) - \gamma(\gamma'^{-1}(e^{g_i}))}{\phi/\omega}\right\}.$$

Given that $\frac{d}{dz}(\gamma'^{-1}(z)) = \frac{1}{\gamma''(\gamma'^{-1}(z))}$, we deduce that

$$\begin{aligned}\frac{d}{dg_i}\gamma\left(\gamma'^{-1}(e^{g_i})\right) &= \gamma'\left(\gamma'^{-1}(e^{g_i})\right)\frac{d}{dg}\gamma'^{-1}(e^{g_i}) \\ &= \frac{e^{2g_i}}{\gamma''(\gamma'^{-1}(e^{g_i}))}\,.\end{aligned}$$

Therefore, the first-order derivative of $\ln p(y_i \mid g_i)$ is equal to

$$\begin{aligned}\frac{d}{dg_i}\ln p(y_i \mid g_i) &= \frac{\omega}{\phi}\left(y_i\frac{d}{dg_i}\gamma'^{-1}(e^{g_i}) - \frac{d}{dg_i}\gamma\left(\gamma'^{-1}(e^{g_i})\right)\right) \\ &= \frac{\omega}{\phi}\left(\frac{y_i\,e^{g_i} - e^{2g_i}}{\gamma''(\gamma'^{-1}(e^{g_i}))}\right),\end{aligned}$$

which becomes (2.8) when rewritten as a vector. Deriving again this last expression leads to

$$\frac{d^2}{dg_i^2}\ln p(y_i \mid g_i) = \frac{\omega}{\phi}\frac{d}{dg_i}\left[\frac{y_i\,e^{g_i} - e^{2g_i}}{\gamma''(\gamma'^{-1}(e^{g_i}))}\right]. \tag{B.1}$$

Given that

$$\frac{d}{dg_i}\left[\frac{1}{\gamma''\left(\gamma'^{-1}\left(e^{g_i}\right)\right)}\right] \;=\; -\frac{e^{g_i}\gamma'''\left(\gamma'^{-1}\left(e^{g_i}\right)\right)}{\left(\gamma''\left(\gamma'^{-1}\left(e^{g_i}\right)\right)\right)^3}\,,$$

a direct calculation allows us to write

$$\begin{aligned}\frac{d}{dg_i}\left[\frac{y_i\,e^{g_i}-e^{2g_i}}{\gamma''\left(\gamma'^{-1}\left(e^{g_i}\right)\right)}\right] &= \frac{y_i\,e^{g_i}-2e^{2g_i}}{\gamma''\left(\gamma'^{-1}\left(e^{g_i}\right)\right)}+\left(y_i\,e^{g_i}-e^{2g_i}\right)\frac{d}{dg}\left[\frac{1}{\gamma''\left(\gamma'^{-1}\left(e^{g_i}\right)\right)}\right] \\ &= \frac{y_i\,e^{g_i}-2e^{2g_i}}{\gamma''\left(\gamma'^{-1}\left(e^{g_i}\right)\right)}-\frac{\left(y_i\,e^{2g_i}-e^{3g_i}\right)\gamma'''\left(\gamma'^{-1}\left(e^{g_i}\right)\right)}{\left(\gamma''\left(\gamma'^{-1}\left(e^{g_i}\right)\right)\right)^3}\,.\end{aligned} \tag{B.2}$$

Combining (B.2) and (B.1), we finally obtain the second-order derivative of $\ln p\left(y_i\,|\,g_i\right)$:

$$\frac{d^2}{dg_i^2}\ln p\left(y_i\,|\,g_i\right)=\frac{\omega}{\phi}\left(\frac{y_i\,e^{g_i}-2e^{2g_i}}{\gamma''\left(\gamma'^{-1}\left(e^{g_i}\right)\right)}-\frac{\left(y_i\,e^{2g_i}-e^{3g_i}\right)\gamma'''\left(\gamma'^{-1}\left(e^{g_i}\right)\right)}{\left(\gamma''\left(\gamma'^{-1}\left(e^{g_i}\right)\right)\right)^3}\right)\,.$$

The matrix $H(\boldsymbol{g})$ in (2.9) is the diagonal matrix of $-\frac{\partial^2}{\partial g_i^2}\ln p\left(y_i\,|\,g_i\right)$. This ends the proof.

C Proof of Proposition 2.3

Without any assumption on the distribution and link function, the fitted value $\hat{y}'$ is obtained from the approximation

$$\begin{aligned}\mathbb{E}\left(Y'\,|\,\mathcal{D},\boldsymbol{x}'\right) &= \int l^{-1}\left(g'\right)\,p\left(g'\,|\,\mathcal{D},\boldsymbol{x}'\right)dg' \\ &\approx \int l^{-1}\left(g'\right)\,q\left(g'\,|\,\mathcal{D},\boldsymbol{x}'\right)dg'\end{aligned} \tag{C.1}$$

where

$$q\left(g'\,|\,\mathcal{D},\boldsymbol{x}'\right)\sim\mathcal{N}\left(\mu_q\left(\boldsymbol{x}'|\mathcal{D}\right)\,,\,\sigma_q^2\left(\boldsymbol{x}'|\mathcal{D}\right)\right)\,. \tag{C.2}$$

Using the properties of the multivariate Normal distribution and (2.12), the expectation $\mu_q\left(\boldsymbol{x}'|\mathcal{D}\right)$ is given by

$$\begin{aligned}\mu_q\left(\boldsymbol{x}'|\mathcal{D}\right) &= \mathbb{E}_q\left(\mathbb{E}\left(G'|\mathcal{D},\boldsymbol{g},\boldsymbol{x}'\right)|\mathcal{D}\right) \\ &= \mathbb{E}_q\left(\mu_g\left(\boldsymbol{x}'|\mathcal{D},\boldsymbol{g}\right)|\mathcal{D}\right) \\ &= k(\boldsymbol{x}',\boldsymbol{X})^\top C(\boldsymbol{X},\boldsymbol{X})^{-1}\hat{\boldsymbol{g}}\,.\end{aligned} \tag{C.3}$$

The variance $\sigma_q^2\left(\boldsymbol{x}'|\mathcal{D}\right)$ can be rewritten as

$$\begin{aligned}\sigma_q^2\left(\boldsymbol{x}'|\mathcal{D}\right) &= \mathbb{V}\left(G'|\mathcal{D},\boldsymbol{x}'\right) \\ &= \mathbb{E}\left(\mathbb{V}\left(G'|\mathcal{D},\boldsymbol{x}',\boldsymbol{g}\right)\right)+\mathbb{V}_q\left(\mathbb{E}\left(G'|\mathcal{D},\boldsymbol{x}',\boldsymbol{g}\right)\right)\,.\end{aligned} \tag{C.4}$$

The first term is provided by (2.13), whereas the second term is equal to

$$\begin{aligned}\mathbb{V}_q\left(\mathbb{E}\left(G'|\mathcal{D},\boldsymbol{x}',\boldsymbol{g}\right)\right) &= \mathbb{V}_q\left(\mu_g\left(\boldsymbol{x}'|\mathcal{D},\boldsymbol{g}\right)\right) \\ &= k(\boldsymbol{X},\boldsymbol{x}')^\top C(\boldsymbol{X},\boldsymbol{X})^{-1}\,\Sigma_{\boldsymbol{g}}\,C(\boldsymbol{X},\boldsymbol{X})^{-1}k(\boldsymbol{X},\boldsymbol{x}')\,.\end{aligned}$$

Inserting this expression in (C.4), we obtain

$$\begin{aligned}\sigma_q^2\left(\boldsymbol{x}'|\mathcal{D}\right) &= k(\boldsymbol{x}',\boldsymbol{x}') - k(\boldsymbol{X},\boldsymbol{x}')^\top\Big[C(\boldsymbol{X},\boldsymbol{X})^{-1} - C(\boldsymbol{X},\boldsymbol{X})^{-1} \\ &\qquad \times\left(H(\hat{\boldsymbol{g}}) + C(\boldsymbol{X},\boldsymbol{X})^{-1}\right)^{-1} C(\boldsymbol{X},\boldsymbol{X})^{-1}\Big]k(\boldsymbol{X},\boldsymbol{x}')\,.\end{aligned}$$

The matrix inversion lemma (see, e.g., Rasmussen and Williaws, 2006, appendix A) states that

$$\left(Z+UWV^\top\right)^{-1} \;=\; Z^{-1} - Z^{-1}U\left(W^{-1}+V^\top Z^{-1}U\right)^{-1}V^\top Z^{-1}$$

where Z is $n \times n$, W is $p \times p$ (with $p \leq n$) and U and V of size $n \times p$. We obtain (2.16) using this lemma with $Z = C(\boldsymbol{X},\boldsymbol{X})$, $W = H(\hat{\boldsymbol{g}})^{-1}$ and U, V that are identity matrices. In the Normal case with canonical link, $\hat{y}'$ is the expectation (2.15) of $q\left(g' \,|\, \mathcal{D}, \boldsymbol{x}'\right)$**.** For the Gamma, Inverse Gaussian, Poisson, Binomial distributions with log-link function, $\hat{y}'$ is the expectation (2.18) of a Log-Normally distributed random variable. This ends the proof.

D Batch K-means

At each iteration, a new random sample of b records is obtained and used to update the clusters in the K-means algorithm, taking care of deprecating previous coordinates according to a learning speed. This operation is repeated until convergence. The procedure is summarized in Algorithm 2.

Algorithm 2 Batch K-means algorithm

```
Initialization:
    Randomly set up initial positions of $K$ centroids
    Initialize clusters $S_1^{(0)} = .... = S_K^{(0)} = \emptyset$
Main procedure:
    For $e = 1$ to maximum epoch, $e_{max}$
        Random sampling of the batch dataset $M$ of size $b$
        Initialize sample clusters $S_1^{(e)} = .... = S_K^{(e)} = \emptyset$
        Assignment step:
        For $i = 1$ to $b$
            1) Assign $i^{th}$ policy to cluster $S_u^{(e)}$ where
               $$S_u^{(e)} = \{u | d(i\,, \boldsymbol{c}_u(e-1)) \le d(i\,, \boldsymbol{c}_j(e-1)) \text{ for } j = 1, ..., K\}\,.$$
        End loop on batch data set, $i$.
        Update step:
        For $u = 1$ to $K$
            2) Calculate the centroids of the batch assigned to $S_u^{(e)}$:
               $$\boldsymbol{c}_u(e) = \frac{1}{|S_u^{(e)}|} \sum_{i \in S_u^{(e)}} \boldsymbol{x}_i\,.$$
            3) Let $\eta_u(e) = \frac{|S_u^{(e)}|}{|S_u^{(e-1)}| + |S_u^{(e)}|}$. Update centroids $\boldsymbol{c}_u(e)$ :
               $$\boldsymbol{c}_u(e) = (1 - \eta_u(e))\, \boldsymbol{c}_u(e-1) + \eta_u(e) \boldsymbol{c}_u(e)\,.$$
        End loop on centroids $u$.
    End loop on epochs $e$
```

Detr²lytics

People drive actuarial innovation

www.detralytics.com - info@detralytics.eu