In this paper we provide a data-augmentation scheme that unifies many common sparse Bayes estimators into a single class. This leads to simple iterative algorithms for estimating the posterior mode under arbitrary combinations of likelihoods and priors within the class. The class itself is quite large: for example, it includes quantile regression, support vector machines, and logistic and multinomial logistic regression, along with the usual ridge regression, lasso, bridge/ellα estimators, and regression with heavy-tailed errors. To arrive at this unified framework, we represent a wide class of objective functions as variance–mean mixtures of Gaussians involving both the likelihood and penalty functions. This generalizes existing theory based solely on variance mixtures for the penalty function, and allows the theory of conditionally normal linear models to be brought to bear on a much wider class of models. We focus on two possible choices of the mixing measures: the generalized inverse-Gaussian and Polya distributions, leading to the hyperbolic and zd distributions, respectively. We exploit this conditional normality to find sparse, regularized estimates using tilted iteratively re-weighted least squares (TIRLS). Finally, we characterize the conditional moments of the latent variances for any model in our proposed class, and show the relationship between our method and two recent algorithms: LQA (local quadratic approximation) and LLA (local linear approximation).

Keywords: variance–mean Gaussian mixtures, sparse regression, classification, data augmentation, quantile regression, support vector machines

1 Regularized regression and classification

1.1 Introduction

In this paper we provide a data-augmentation scheme that unifies a wide variety of procedures for regularized regression and classification into a single class. The main practical result of this unification is to suggest a simple, all-purpose algorithm for estimating the posterior mode in many non-Gaussian problems that, up to now, have required tailored or approximate methods. This algorithm (which we call tilted, iteratively re-weighted least squares) is efficient, straightforward to implement, and easily parallelized to exploit a multi-core computing environment. Moreover, it can be implemented in a way that avoids both matrix inversion and numerical differentiation.

Our unified class includes many well-known likelihoods and priors, in arbitrary combinations: quantile regression, support vector machines, penalized logistic and multinomial logistic regression, ridge regression, lasso, bridge estimators, topic models, autologistic models, and penalized regression with heavy-tailed errors. It also includes many techniques that are widely used in machine learning, such as mixtures of logits, restricted Boltzmann machines, and multi-layer neural networks.

In all of these problems, maximum a posteriori (MAP) estimation entails minimizing an ob-
Objective function of the form

\[ Q(\beta) = \sum_{i=1}^{n} f(y_i, x_i^T \beta) + \sum_{j=1}^{p} g\left(\frac{\beta_j}{\tau s_j}\right). \]

Here \( f \) and \( g \) are specified functions describing the log likelihood and log prior (or penalty); \( y_i \) is a response, which may be continuous or multinomial; \( x_i \) is a \( p \)-vector of predictors; \( \beta = (\beta_1, \ldots, \beta_p) \) is a vector of predictor loadings; \( \tau \) controls the strength of regularization; and the \( \{s_j\} \) are fixed scale factors, often equal to 1 and always specified in advance.

Our work is motivated by recent Bayesian research on so-called “sparsity priors” in normal linear regression, where \( f \) is the sum of squared residuals and \( g \) is the log of some normal variance-mixture prior having favorable properties for estimating sparse signals. Recent examples of this work include [Figueiredo, 2003], [Bae and Mallick, 2004], [Griffin and Brown, 2005], [Park and Casella, 2008], [Hans, 2009], [Carvalho et al., 2010], [Griffin and Brown, 2010], and [Armagan et al., 2010].

In generalizing this line of work, we use the theory of hierarchical variance–mean mixtures of Gaussians to represent both the penalty \( f \) and the likelihood \( g \). In this way many common nonconcave penalized-likelihood problems can be recast as conditionally normal linear models under a conditionally normal prior. This allows the original problem to be solved using a variation on iteratively re-weighted least squares, exploiting standard Bayesian linear-model theory under conjugate priors [Lindley and Smith, 1972].

Our data-augmentation approach offers a number of advantages. The first is its sheer simplicity, best illustrated by an example involving familiar choices of \( f \) and \( g \). Suppose we wish to fit a logistic regression with a bridge penalty, where

\[ \hat{\beta} = \arg \min_{\beta \in \mathbb{R}^p} \left\{ \sum_{i=1}^{n} \log\left(1 + \exp\{-y_i x_i^T \beta\}\right) + \sum_{j=1}^{p} |\beta_j / \tau s_j|^a \right\}, \]

assuming that the outcomes \( y_i \) are coded as \( \pm 1 \). Many factors conspire to make this a difficult problem: the parameter vector is high-dimensional, likelihood is non-Gaussian, and the log-prior leads to a non-convex constraint set. But the results in this paper (specifically, those of Section 2) show that \( \hat{\beta} \) can be found with minimal computational fuss by starting with initial guesses \( \{\beta(0), \omega(0), \lambda(0)\} \) and iterating the following three steps until convergence:

\[
\begin{align*}
\beta^{(s+1)} &= \left(\tau^{-2} S^{-1} \Lambda^{-1(\Sigma)} + X_\Sigma^T \hat{\Omega}^{-1(\Sigma)} X_\Sigma\right)^{-1} \left(\frac{1}{2} X_i^T 1\right), \\
\omega_j^{(s+1)} &= \frac{1}{z_j^{(s)}} \left\{ \frac{e^{-z_j^{(s)}}}{1 + e^{-z_j^{(s)}}} - \frac{1}{2} \right\}, \\
\lambda_j^{(s+1)} &= \alpha (\tau s_j)^{2-\alpha} |\beta_j^{(s)}|^{\alpha-2},
\end{align*}
\]

where \( z_j^{(s)} = y_j x_j^T \beta(\Sigma) \); \( X_\Sigma \) is the matrix having rows \( x_i^T = y_i x_i \); \( 1 \) is a vector of ones; and where \( \Omega = \text{diag}(\omega_1, \ldots, \omega_n) \), \( \Lambda = \text{diag}(\lambda_1, \ldots, \lambda_p) \), and \( S = \text{diag}(s_1, \ldots, s_p) \) are diagonal matrices. The data-augmentation variables \( \{\omega_i\} \) and \( \{\lambda_j\} \) are used to derive a conditionally normal representation of
the objective function, avoiding the use of tailored methods for specific combinations of likelihood and penalty [e.g. Huang et al., 2008; Taddy, 2010].

The second main advantage of our approach is its generality. With only slight modifications of this iterative scheme, solutions can be found to all of the other regularization problems mentioned above. Other authors have proposed hybrid algorithms that represent particular non-concave penalty functions \( g \) as scale mixtures of normals, and then use versions of EM to find the minimum [e.g. Figueiredo, 2003; Armagan et al., 2010]. These algorithms offer alternatives to LARS [Efron et al., 2004] tailored to the case where \( f(z_i \mid \beta) \) corresponds to a Gaussian likelihood.

Our approach, on the other hand, allows for the same algorithm to be used for finding the solution across a broad range of problems, even when arbitrarily “mixing and matching” likelihood and penalty terms within the proposed class—for example, a double-Pareto penalty with a logistic likelihood, or quantile regression [Koenker, 2005] with \( \ell^\alpha \) regularization. We call this algorithm tilted, iteratively re-weighted least squares, or TIRLS. Different likelihoods correspond to different updates for the latent \( \omega_i \)'s, while different penalties correspond to different updates for the latent \( \lambda_j \)'s. The general utility of the algorithm for non-Gaussian likelihoods is of particular interest, given that the computational advantages of the LARS algorithm derive largely from the squared-error loss function.

In this respect, our Theorem 3.2 is crucial: it shows that there is a simple relationship between the derivatives of \( f \) and \( g \) and the updates (or E step) for the augmentation variables \( \omega_i \) and \( \lambda_j \). These updates can usually be calculated in closed form, even if the full conditional distribution of the augmented variables is unknown or intractable. In addition, this theorem relates our approach to two other algorithms for fitting regularized estimators using local approximations to the penalty function: LQA [Fan and Li, 2001] and LLA [Zou and Li, 2008]. It is known that these algorithms are exact EM algorithms for penalty functions that are normal scale mixtures. We provide a much more general approach involving non-Gaussian likelihoods.

Theorem 3.3 provides the corresponding result for the posterior mean estimator. It provides a generalization of the commonly usedMasreliez [1975] theorem in robust Bayesian statistics, generalizing the result of Pericchi and Smith [1992].

The final advantage of our approach is its parallelizability. Our algorithm can be sped up by a factor that is very nearly linear in the number of processor cores available. This lends further scalability to the method, and constrasts sharply with many other general-purpose algorithms for estimating sparse non-Gaussian models—most notably coordinate descent, which cannot be parallelized even in principle.

1.2 Relationship with previous work

Finding the estimator \( \hat{\beta} = \arg \min_{\beta \in \mathbb{R}^p} Q(\beta) \) in our data-augmentation approach relies upon representing the solution to (1) as a MAP estimator \( \hat{\beta} \) for \( p(\beta \mid \tau, y) \propto e^{-Q(\beta)} \). Specifically, we
have the following decomposition

\[
e^{-Q(\beta)} \propto p(\beta \mid \tau, y) \propto \exp \left\{ -\sum_{i=1}^{n} f(y_i, x_i^T \beta) - \sum_{j=1}^{p} g(\beta_j / \tau z_j) \right\}
\]

\[
\propto \left\{ \prod_{i=1}^{n} p(z_i \mid x_i^T \beta) \right\} \prod_{j=1}^{k} p(\beta_j \mid \tau)
\]

\[
= p(z \mid \beta) \cdot p(\beta \mid \tau),
\]

where \( z_i = y_i - x_i^T \beta \) for regression, or \( z_i = y_i x_i^T \beta \) for classification (with the response \( y_i \) coded as \( \pm 1 \)). Although we briefly explore fully Bayesian approaches for working with this high-dimensional joint posterior distribution, for the most part we concentrate on finding the pseudo-posterior mode as the solution to the original regularization problem.

Within this class of regularized estimators, there has been widespread interest in cases where the penalty function \( g \) corresponds to a normal variance mixture. This subclass includes many estimators that enjoy broad use, and that have been studied in detail from both classical and Bayesian perspectives. Some examples include the lasso [Tibshirani, 1996; Park and Casella, 2008; Hans, 2009], bridge estimators [West, 1987; Huang et al., 2008], the relevance vector machine of Tipping [2001], the normal/Jeffreys model of Figueiredo [2003] and Bae and Mallick [2004], the normal/exponential-gamma model of Griffin and Brown [2005], the normal/gamma and normal/inverse-Gaussian [Caron and Doucet, 2008; Griffin and Brown, 2010], the horseshoe prior of Carvalho et al. [2010], the hypergeometric inverted–beta model of Polson and Scott [2010b], and the double-Pareto model of Armagan et al. [2010].

Our paper extends this literature in three ways. First, we show that for a very wide class of regularized estimators, both the pseudo-likelihood \( p(z_i \mid \beta) \) and the prior/penalty \( p(\beta_j \mid \tau) \) can be represented as variance–mean mixtures of normals:

\[
p(z_i \mid \beta) = \int_{0}^{\infty} \phi(z_i \mid \mu_z + \kappa_z \omega_i, \sigma^2 \omega_i) \, dP(\omega_i)
\]

\[
p(\beta_j \mid \tau) = \int_{0}^{\infty} \phi(\beta_j \mid \mu_\beta + \kappa_\beta \lambda_j, \tau^2 s_j^2 \lambda_j) \, dP(\lambda_j).
\]

Here \( \phi(a \mid m, v) \) denotes the density function of the normal distribution having mean \( m \) and variance \( v \), evaluated at \( a \).

We show that, by choosing different fixed combinations of \( (\mu_z, \kappa_z, \mu_\beta, \kappa_\beta) \) and the mixing distributions \( p(\omega_i) \) and \( p(\lambda_j) \), it is possible to generate many commonly used objective functions after marginalizing out the data augmentation variables \( \{\omega_i\} \) and \( \{\lambda_j\} \). Given these latent variances, both the loss and penalty functions are conditionally normal, reducing the problem to a linear model with heteroscedastic errors.

This generalizes recent work on regularization, in that we employ variance–mean mixtures rather than simply variance mixtures, and we use a mixture representation for both the pseudo-likelihood and the prior. The hyperparameter of the pseudo-prior distribution, \( \tau \), is usually conceived as a tuning parameter governing the strength of regularization. It maps one-to-one with the global scale parameter for \( p(\beta_j \mid \lambda_j, \tau) \) in our variance–mean mixture framework.
This class nests all procedures corresponding to variance mixtures of normals, such as the lasso and bridge estimators. But it is much larger. Indeed, as we mentioned in the introduction and will show in detail in Section 2, the latent-variable approach allows the theory of normal linear models to be brought on bear on a very broad range of common regularization procedures. For example, our new class provides a latent-variable representation for regularized versions of support-vector machines; quantile regression; and logistic regression for binary, ordinal, and multinomial outcomes. This facilitates interpretation and leads to greater ease in building predictive models that incorporate uncertainty in forecasting, a point raised by Hans [2010] in the context of the lasso estimator. Previous studies [e.g. Polson and Scott, 2011b] have presented a similar result for specific models, but as far as we are aware, ours is the first characterization of the full class.

Second, we describe a tilted, iteratively re-weighted least squares (TIRLS) algorithm for finding solutions to any regularization problem in this class. By exploiting the fact that \( \hat{\beta} \) is the pseudo-posterior mode under a high-dimensional mixture of normals with latent variances \( \{\omega_i\} \) and \( \{\lambda_j\} \), we show that any such regularized estimator can be found using a slight variation of iteratively re-weighted least squares. These algorithms, described in Section 3, provide an alternative to standard numerical optimization routines; provide a general solution even for non-convex penalty functions; and are general enough to provide an exact analysis for many problems (including topic models in natural-language processing) that have hitherto been been solved using variational-Bayes methods. They also extend naturally to full MCMC algorithms, if one wishes to compute a full posterior distribution for \( \hat{\beta} \); see, for example, the MCMC approaches for non-Gaussian likelihoods in Carlin and Polson [1991] and Gramacy and Polson [2010].

There are other algorithms leading to penalized-likelihood solutions for more general penalty functions \( g(\beta_j) \) than can be accommodated within our framework. For example, the SCAD penalty [Fan and Li, 2001] has no variance–mean mixture representation, yet can be fit with the LLA algorithm described in Zou and Li [2008]. But the generality of our approach comes from interweaving two mixture representations, one for the likelihood and one for the penalty function. This naturally leads to sparse estimators for many different non-Gaussian likelihoods.

Third, we prove a theorem characterizing the conditional posterior moments of the latent variables \( \{\omega_i\} \) and \( \{\lambda_j\} \). These moments are necessary inputs in our algorithm. Previous authors have derived analytic expressions for these moments in special cases, facilitating EM-style algorithms; see, for example, Armagan et al. [2010]. Our theorem allows these moments to be calculated in closed form for any combination of error model and penalty term in the proposed class, even when the full conditional posterior distribution of the latent variables is unknown or analytically intractable.

2 A latent-variable representation for a wide class of regularized estimators

2.1 Normal variance–mean mixtures

We will show that, by introducing data-augmentation variables \( \{\omega_i\} \) and \( \{\lambda_j\} \), many commonly used loss functions and regularization penalties can be expressed in form (3–4), with different
Table 1: A selection of variance-mean mixture representations corresponding to many common loss and penalty functions. GIG: generalized inverse-Gaussian.

<table>
<thead>
<tr>
<th>Error/loss function</th>
<th>$f(z_i \mid \beta)$</th>
<th>$\kappa_z$</th>
<th>$\mu_z$</th>
<th>$p(\omega_i)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Squared-error</td>
<td>$z_i^2$</td>
<td>0</td>
<td>0</td>
<td>$\omega_i \equiv 1$</td>
</tr>
<tr>
<td>Absolute-error</td>
<td>$</td>
<td>z_i</td>
<td>$</td>
<td>0</td>
</tr>
<tr>
<td>Check loss</td>
<td>$</td>
<td>z_i</td>
<td>+ (2q - 1)</td>
<td>z_i</td>
</tr>
<tr>
<td>Support vector machines</td>
<td>$\max(1 - z_i, 0)$</td>
<td>1</td>
<td>1</td>
<td>GIG</td>
</tr>
<tr>
<td>Logistic</td>
<td>$\log(1 + e^{</td>
<td>\beta</td>
<td>})$</td>
<td>1/2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Penalty function</th>
<th>$g(\beta_j \mid \tau)$</th>
<th>$\kappa_\beta$</th>
<th>$\mu_\beta$</th>
<th>$p(\lambda_j)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ridge</td>
<td>$(\beta_j / \tau)^2$</td>
<td>0</td>
<td>0</td>
<td>$\omega_i \equiv 1$</td>
</tr>
<tr>
<td>Lasso</td>
<td>$</td>
<td>\beta_j</td>
<td>/ \tau$</td>
<td>0</td>
</tr>
<tr>
<td>Bridge</td>
<td>$</td>
<td>\beta_j</td>
<td>^\alpha$</td>
<td>0</td>
</tr>
<tr>
<td>Generalized Double-Pareto</td>
<td>${(1 + \alpha) / \tau} \log(1 +</td>
<td>\beta_j</td>
<td>/ \alpha \tau)$</td>
<td>0</td>
</tr>
</tbody>
</table>

fixed choices of the parameters and mixing measures:

$$p(z_i \mid \beta) = \int_0^\infty \phi(z_i \mid \mu_z + \kappa_z \omega_i, \sigma^2 \omega_i) \, dP(\omega_i)$$

$$p(\beta_j \mid \tau) = \int_0^\infty \phi(\beta_j \mid \mu_\beta + \kappa_\beta \lambda_j, \tau^2 \lambda_j^2) \, dP(\lambda_j).$$

Table 1 lists several common objective functions, along with the corresponding choices for $(\kappa_\beta, \mu_\beta)$, $(\kappa_z, \mu_z)$, and the mixing distributions.

An important feature of our approach is that we avoid dealing directly with conditional distributions for these latent variables. To find the mode, it is sufficient merely to use Theorem 3 to calculate the appropriate moments of these distributions. These moments depend only upon the derivatives of $f$ and $g$, along with the hyperparameters. From a practitioner’s perspective, this is the key step in implementing our TIRLS algorithm, described in Section 3.

Any choice of the mixing measures $P(\omega_i)$ and $P(\lambda_j)$ in (3) and (4) will lead to a marginal pseudo-posterior whose mode is a regularized estimator in our class. But we focus on two choices in particular: the generalized inverse-Gaussian (GIG) distribution, and the Polya distribution. These two choices lead to the hyperbolic and Z distributions, respectively, for the resulting variance-mean mixture.

The two key integral identities for the hyperbolic/GIG and Z/Polya mixtures are

$$\frac{\alpha^2 - \kappa^2}{2\alpha} e^{-\alpha|\theta - \mu| + \kappa(\theta - \mu)} = \int_0^\infty \phi(\theta \mid \mu + \kappa v, v) \, p_{\text{GIG}}(v \mid 1, 0, \sqrt{\alpha^2 - \kappa^2}) \, dv$$  (5)

$$\frac{1}{B(\alpha, \kappa)} \frac{e^{\alpha(\theta - \mu)}}{(1 + e^{\theta - \mu})^{2(\alpha - \kappa)}} = \int_0^\infty \phi(\theta \mid \mu + \kappa v, v) \, p_{\text{Polya}}(v \mid \alpha, \alpha - 2\kappa) \, dv.$$  (6)

These two expressions lead, in turn, to three further identities concerning the improper limits of
GIG and Polya mixing measures for variance–mean Gaussian mixtures:

\[
a^{-1} \exp \left\{ -2c^{-1} \max(a\theta, 0) \right\} = \int_0^\infty \phi(\theta \mid -av, cv) \, dv
\]

\[
e^{-1} \exp \left\{ -2c^{-1} \rho_\theta(\theta) \right\} = \int_0^\infty \phi(\theta \mid -(2\tau - 1)v, cv)e^{-2\tau(1-\tau)v} \, dv
\]

\[
(1 + \exp\{\theta - \mu\})^{-1} = \int_0^\infty \phi(\theta \mid \mu - (1/2)v, v) \, p_{PY}(v \mid 0, 1) \, dv
\]

where \( \rho_\theta(\theta) = \frac{1}{2}|\theta| + \left( q - \frac{1}{2} \right) \theta \) is the check-loss function \cite{johannes2009, li2010}. The first relates to support-vector machines (SVM); the second, to quantile and lasso regression; and the third, to logistic and multinomial logistic regression. The function \( p_{PY}(\lambda) \) is an improper measure given by a sum of exponentials \cite{gramacy2010}. With GIG and Polya mixing distributions alone, one can generate the following objective functions:

\[
u^2, \max(u, 0), |u| + (2\tau - 1)u, \frac{1}{1 + e^{-u}}, \text{ and } \frac{1}{(1 + e^{-u})^\delta}.
\]

These correspond to the ridge, lasso, support-vector machine, check loss/quantile regression, logit, and multinomial models, respectively. More general families can generate other penalty functions—for example, the bridge penalty \(|u|^\delta\) from a stable mixing distribution \cite{west1987}.

### 2.2 Generalized hyperbolic distributions

In all of the following cases, we assume that \((\theta \mid v) \sim \mathcal{N}(\mu + \kappa v, v)\), and that \(v \sim p(v)\). Let \(p(v)\) be a generalized inverse-Gaussian distribution \(GIG(\psi, \gamma, \delta)\); details of this family are given in the appendix. We consider the special case of this class where \(\psi = 1\) and \(\delta = 0\), in which case \(p(\theta)\) is a hyperbolic distribution having density function

\[
p(\theta \mid \mu, \gamma) = \left( \frac{\alpha^2 - \kappa^2}{2\alpha} \right) \exp \left\{ -\alpha|\theta - \mu| + \kappa(\theta - \mu) \right\}.
\]

When viewed as a pseudo-likelihood or pseudo-prior, the class of generalized hyperbolic distributions will generate the following penalty functions.

**Lasso:** choosing \((\alpha, \gamma, \mu) = (1, 0, 0)\) leads to \(- \log p(\beta_j) = |\beta_j|\), corresponding to \(\ell^1\) regularization.

**Check loss/quantile regression:** choosing \((\alpha, \gamma, \mu) = (1, 1 - 2\gamma, 0)\) leads to

\[- \log p(z_i) = |z_i| + (2\gamma - 1)z_i.
\]

which leads to quantile regression for the \(q\)th quantile.

**Support vector machines:** choosing \((\alpha, \gamma, \mu) = (1, 1, 1)\) leads to

\[-(1/2) \log p(z_i) = (1/2)|1 - z_i| + (1/2)(1 - z_i) = \max(1 - z_i, 0)
\]
for $z_i = y_i x_i^T \beta$. This is the objective function for classification using support vector machines. It corresponds to the limiting case of a generalized inverse-Gaussian prior, but is still a valid choice of hyperparameters in light of the fact the identities summarized above. See Polson and Scott [2011b]. Under the limiting result that

$$(\alpha^2 - \kappa^2)^{-1} p_{GIG(1,0,\sqrt{\alpha^2-\kappa^2})}(\lambda) \equiv 1$$

when $\alpha$ and $\kappa$ take on the same value, it is as if we have a uniform prior, $p(\omega_i) \propto 1$.

Other members of this class include familiar forms such as the Student $t$, Cauchy, and normal–Gamma distributions, in addition to other variance–mean mixture generalizations that have been used in finance. The GIG mixture is a special case (with $\alpha = \frac{1}{2}$) of a modified normal-stable process with penalty $\int_0^\infty \phi(\theta | \mu + \kappa v, v) \ p_{MS}(v | a, v, \gamma, \delta) \ dv$, see [Barndorff-Nielsen and Shephard, 2001] for density definitions. Another member of this class is the tempered stable distribution which leads to the bridge estimator with penalty $-\log p(\beta_j) = |\beta_j|^\alpha$.

### 2.3 Z distributions

Now let $p_{\polya}(v | a, a - 2\kappa)$ be a Polya distribution, which can be represented as an infinite convolution of exponentials. Details of this distribution are presented in the appendix.

A Polya mixing distribution for the exchangeable random variances $\omega_i$ can be used to generate the class of $Z$ distributions. The important result is the following:

$$p_Z(\theta | \mu, a, \kappa) = \frac{1}{B(a, \kappa)} \left( e^{\theta - \mu} \right)^a = \int_0^\infty \mathcal{N}(\mu + \kappa v, v) p_{\polya}(v | a, a - 2\kappa) \ dv.$$  

See Barndorff-Nielsen et al. [1982]. When viewed as a pseudo-likelihood, the class of Polya/Z distributions results in the logistic and multinomial models.

**Logit:** choosing $(a, \kappa, \mu) = (1, 1/2, 0)$ leads to

$$p(z_i) = \frac{e^{\eta_i}}{1 + e^{\eta_i}},$$

which is the likelihood for logistic regression with $z_i = y_i x_i^T \beta$. Much like the support vector-machine representation, this corresponds to a limiting improper case of the Polya distribution, specifically $\mathcal{P} \mathcal{Y}(1,0)$. The necessary mixture representation still holds, however, in light of the integral identities presented above. The improper mixing measure $p_{1,0}(v)$ is an infinite sum of exponentials for which the integral on the right still converges to the logistic likelihood [Gramacy and Polson, 2010].

**Multinomial:** For the multinomial generalization of the logistic model, we require a slight modification. Suppose that $y_i \in \{1, \ldots, K\}$ is an unordered category indicator, and that $\beta_k = (\beta_{k1}, \ldots, \beta_{kp})^T$ is a separate set of $p$ regression coefficients for the $k$th category. Let

$$\eta_{ij} = \exp \left( x_i^T \beta_j - c_{ij} \right) / \{1 + \exp \left( x_i^T \beta_j - c_{ij} \right) \},$$

then

$$p(z_{i1}, \ldots, z_{iK}) = \prod_{k=1}^K \frac{e^{\eta_{ik}}}{1 + e^{\eta_{ik}}},$$

and

$$p(z_i) = \frac{1}{1 + e^{\eta_i}}.$$
where \( c_{ij}(\beta_{-j}) = \ln \sum_{k \neq j} \exp \{ x_i^T \beta_k \} \).

We follow Holmes and Held [2006] in writing the conditional likelihood as

\[
Q(\beta_j | \beta_{(-j)1}, y) \propto \prod_{i=1}^{n} \prod_{k=1}^{K} \eta_{ik}^{I(y_i = k)} \\
\approx \prod_{i=1}^{n} \prod_{k=1}^{K} \eta_{ik}^{I(y_i = j)} \left\{ w_i (1 - \eta_{ij}) \right\}^{I(y_i \neq j)} \\
\approx \prod_{i=1}^{n} \left\{ \frac{\exp \left( x_i^T \beta_j - c_{ij} \right)}{1 + \exp \left( x_i^T \beta_j - c_{ij} \right)} \right\},
\]

where \( w_i \) is independent of \( \beta_j \) and \( \mathbb{1} \) is the indicator function. Therefore we have a Polya mixture representation for the conditional likelihood, where \( \alpha_{ij} = \mathbb{1}(y_i = j) \), \( \kappa_{ij} = \mathbb{1}(y_i = j) - 1/2 \), and \( \mu_{ij} = c_{ij}(\beta_{-j}) \).

### 3 Data augmentation algorithms

We have shown that many common regularization procedures take the form

\[
z_i = \mu_z + \kappa_z \omega_i + \sigma \sqrt{\omega_i} \epsilon_i \quad \text{with} \quad \omega_i \sim p(\omega_i), \quad (10) \\
\beta_j = \mu_\beta + \kappa_\beta \lambda_j + \tau s_j \sqrt{\lambda_j \eta_j} \quad \text{with} \quad \lambda_j \sim p(\lambda_j). \quad (11)
\]

Here \( s_j \) are known scaling variables, while \( \epsilon_i \) and \( \eta_j \) are mutually independent, standard-normal random variables.

We now show how this conditionally normal representation can be exploited to build an IRLS-style algorithm for computing regularized estimators for any model in the class of normal variance–mean mixtures.

#### 3.1 Tilted, iteratively re-weighted least squares

Our approach involves a variation on the EM algorithm [Dempster et al., 1977], finding \( \hat{\beta} \) by alternating between an E-step (expectation) and an M-step (maximization), iterating until convergence.

**E step:** Compute the expected value of the log posterior, given the current parameter estimate:

\[
Q(\beta | \beta^{(s)}) = \int \log p(\beta | \omega, \lambda, \tau, y) p(\omega, \lambda | \beta^{(s)}, \tau, y) \, d\omega \, d\lambda.
\]

**M step:** Maximize the complete-data posterior to update the parameter estimate:

\[
\beta^{(s+1)} = \arg \max_{\beta} Q(\beta | \beta^{(s)}).
\]

The sequence of estimated parameter values \( \{ \beta^{(1)}, \beta^{(2)}, \ldots \} \) monotonically increases the observed-data objective function.
where under conditional independence, the prior is
\[ c \]
for some constant \( m \). The complete-data log posterior can be computed as follows. Under a normal variance–mean mixture, we have a model of the form
\[ (\beta_j^{(s)} - \mu_{\beta}) \hat{\lambda}_j^{-1(s)} = \kappa_{\beta} + \tau^2 s_j^2 g'(\beta_j^{(s)} | \tau), \]
\[ (z_i^{(s)} - \mu_z) \hat{\omega}_j^{-1(s)} = \kappa_z + \sigma^2 f'(z_i^{(s)} | \beta). \]

If \( \lambda_j^{-1} > \lambda_{\text{max}}^{-1} \), delete \( \lambda_j \) and \( \beta_j \) from the model.

M-Step (\( \mu_\beta = \kappa_\beta = 0 \)) For regression, compute \( \beta^{(s+1)} \) as
\[ \beta^{(s+1)} = \left( \tau^{-2} s_j^{-1} \hat{\lambda}_j^{-1(s)} + X_i^T \hat{\Omega}_j^{-1(s)} X_i \right)^{-1} X_i^T \left( \hat{\Omega}_j^{-1(s)} y - \mu_z \omega_j^{-1(s)} - \kappa_z \right). \]

For classification, compute \( \beta^{(s+1)} \) as
\[ \beta^{(s+1)} = \left( \tau^{-2} s_j^{-1} \hat{\lambda}_j^{-1(s)} + X_i^T \hat{\Omega}_j^{-1(s)} X_i \right)^{-1} X_i^T \hat{\Omega}_j^{-1(s)} \left( \mu_z \hat{1} + \kappa_z \hat{\omega}_j^{(s)} \right). \]

Figure 1: An EM algorithm based on mixtures of ridge regressions for solving a wide variety of regularization problems.

The complete-data log posterior can be computed as follows. Under a normal variance–mean mixture, we have a model of the form
\[ p(\beta | \tau, y) = \int \pi(\beta | \omega, \lambda, y) p(\omega, \lambda | y, \tau) \, d\omega \, d\lambda, \]
where under conditional independence, the prior is
\[ p(\omega, \lambda | \tau) = \prod_{j=1}^p p(\lambda_j | \tau) \cdot \prod_{i=1}^n p(\omega_i | y, \tau). \]

Therefore
\[ \log p(\beta | \omega, \lambda, \tau, y) = c_0(\omega, \lambda, y, \tau) - \frac{1}{2} \sum_{i=1}^n \omega_i^{-1} (z_i - \mu_z - \kappa_y \omega_i)^2 \]
\[ - \frac{1}{2 s_j^2 \tau^2} \sum_{j=1}^p \lambda_j^{-1} (\beta_j - \mu_\beta - \kappa_\beta \lambda_j)^2 \]
for some constant \( c_0 \) and known scaling factors \( s_j \). Through further factorization of this expression in terms of \( \beta \), this can be shown to depend only upon the conditional moments \( \{ \hat{\omega}_j^{-1} \} \) and \( \{ \hat{\lambda}_j^{-1} \} \).

Therefore, to perform the E-step, we calculate the criterion function \( Q(\beta | \beta^{(s)}) \) by simply replacing \( \lambda_j^{-1} \) and \( \omega_j^{-1} \) with their conditional expectations \( \hat{\lambda}_j^{-1(s)} \) and \( \hat{\omega}_j^{-1(s)} \), given the observed data and the current \( \beta^{(s)} \). We postpone a discussion of this step to the following section, where
we prove a theorem that allows these moments to be calculated in closed form for any normal variance–mean mixture where the pseudo-likelihood and penalty functions are known.

The M-step involves computing the posterior mode under a conditionally Gaussian prior for \( \beta \) and a heteroscedastic Gaussian error model, given the current estimates values for the latent variances \( \{ \omega^{-1} \} \) and \( \{ \lambda^{-1} \} \). For regression problems, simple manipulations of (10) and (11) show that the mode depends only upon the conditional moments \( \hat{\omega}^{-1} \) and \( \hat{\lambda}^{-1} \), as required. Only a slight modification is required for a classification problem. But in either case, the conditional maximum is recognizable as a generalized ridge estimator [Denison and George, 2000; Polson and Scott, 2010a], a result which we formalize as follows.

**Theorem 3.1.** Suppose that the objective function \( Q(\beta) \) can be represented by a hierarchical variance–mean Gaussian mixture, as in Equations (3) and (4). Then given estimates \( \{ \hat{\omega}^{-1}_i \} \) and \( \{ \hat{\lambda}^{-1}_i \} \), we have the following expressions for the conditional maximum of \( \beta \), where \( \omega^{-1} = (\omega_1^{-1}, \ldots, \omega_n^{-1})^T \) and \( \lambda^{-1} = (\lambda_1^{-1}, \ldots, \lambda_p^{-1})^T \) are column vectors; and where \( \Omega^{-1} = \text{diag}(\omega_1^{-1}, \ldots, \omega_n^{-1}), \Lambda^{-1} = \text{diag}(\lambda_1^{-1}, \ldots, \lambda_p^{-1}), \) and \( S^{-1} = \text{diag}(\sigma_1^{-1}, \ldots, \sigma_p^{-1}) \) are diagonal matrices.

(A) In a regression problem,

\[
\hat{\beta} = (\tau^{-2}S^{-1}\hat{\Lambda}^{-1}(\beta) + X^T\hat{\Omega}^{-1}(\beta)X)^{-1}(y^* + b^*),
\]

where

\[
y^* = X^T(\hat{\Omega}^{-1}(\beta)y - \mu_\omega \omega^{-1}(\beta) - \kappa_\varepsilon 1),
\]

\[
b^* = (\tau^{-2}S^{-1})(\mu_\beta \lambda^{-1} + \kappa_\beta 1).
\]

(B) In a classification problem where \( y_i = \pm 1 \) and \( X_j \) has rows \( x_j^* = y_i x_i, \)

\[
\hat{\beta} = (\tau^{-2}S^{-1}\hat{\Lambda}^{-1}(\beta) + X_j^T\hat{\Omega}^{-1}(\beta)X_j)^{-1}X_j^T(\mu_\omega \omega^{-1}(\beta) + \kappa_\varepsilon 1).
\]

**Proof.** In appendix. \( \square \)

3.2 Calculating the conditional moments of the latent variables

The following key theorem provides the necessary conditional moments for \( \omega_i \) and \( \lambda_j \) under any model where both the pseudo-likelihood and pseudo-prior can be represented by normal variance–mean mixtures.

**Theorem 3.2.** Suppose that the objective function \( Q(\beta) \) can be represented by a hierarchical variance–mean Gaussian mixture, as in Equations (3) and (4). Then the conditional moments \( \hat{\lambda}_j^{-1}(\beta) \) and \( \hat{\omega}_i^{-1}(\beta) \) are given by the following expressions:

\[
(\hat{\beta}_j - \mu_\beta)\hat{\lambda}_j^{-1}(\beta) = \kappa_\beta + \tau^2 s^2 \ g'(\hat{\beta}_j \mid \tau),
\]

\[
(z_i - \mu_z)\hat{\omega}_i^{-1}(\beta) = \kappa_z + \sigma^2 \ f'(z_i \mid \beta),
\]

where \( f' \) and \( g' \) are the derivatives of the loss and penalty functions from (1), respectively.
The advantage of the theorem is that it characterizes the conditional moments purely in terms of the loss and penalty functions

\[ f(z_i) = -\log p(z_i | \beta) \quad \text{and} \quad g(\beta_j) = -\log p(\beta_j | \tau), \]

which are pre-specified in most regularization problems. This leads to the simple EM-style algorithm based on mixtures of ridge regressions, summarized in the introduction.

One caveat is that for \( \beta_j - \mu_\beta = 0 \), the conditional moment in the \( E \) step may be infinite, and care must be taken. It is important to emphasize that, in TIRLS, numerically infinite values for \( \hat{\lambda}_j^{-1} \) do not reflect a pathology, but rather are the result of a normally functioning algorithm that has found a sparse solution. The numerical difficulty this poses can be easily handled by starting the algorithm from a value where \( \beta - \mu_\beta \) has no zeros, and then removing \( \beta_j \) from the model when it gets within a small numerical threshold of its mean. This conveys the added benefit of hastening the matrix computations in the weighted least-squares (\( M \)) step. Although we have found this approach to work well in practice, it has the disadvantage that a variable cannot re-enter the model once it has been deleted. An alternate approach involves the use of restricted least-squares; for details, see Section 3.2 of [Polson and Scott 2011b].

Our approach is related to that of [Fan and Li 2001], who propose iteratively, locally approximating the penalty function \( g \) using a second-order Taylor expansion, and refer to this as local quadratic approximation (LQA). This corresponds to the iterative scheme

\[
\beta^{(s+1)} = \arg\max \left\{ \sum_{i=1}^{n} (y_i - x_i^T \beta)^2 - \frac{1}{2} \sum_{j=1}^{p} g'(|\beta_j^{(s)}|) |\beta_j^{(s)}|^2 \right\},
\]

where the quadratic approximation to \( g \) is updated at each step to reflect the previous estimate of \( \beta \). This results in a tractable optimization problem.

To accommodate alternatives to the squared-error log-likelihood, Fan and Li suggest using an analogous approximation to \( f \). This approximation will be poor when some of the residuals \( |y_i - x_i^T \beta| \) are small. Our results show that no such approximation is necessary for likelihoods in our proposed class. Moreover, these results also show that LQA is exact in the special case of a penalty function that is a variance mixture.

**Proposition 3.1.** LQA is an exact EM algorithm whenever the penalty function \( g \) can be represented as a normal variance mixture.

The proof follows trivially from the fact that \( \mathbb{E}(\lambda_j^{-1} | \beta) = g'(|\beta_j|) / |\beta_j| \) for a variance-mixture penalty function. The TIRLS update for \( \beta \) is therefore the solution to the same minimization problem as LQA, to wit:

\[
\beta^{(s+1)} = \arg\max \left\{ \sum_{i=1}^{n} f(z_i) - \frac{1}{2} \sum_{j=1}^{p} \beta_j^{s^2} / \lambda_j \right\}.
\]

[Zou and Li 2008] propose an alternative approach called LLA, which uses a local linear approximation to the penalty function \( g \). This involves canceling a factor of \( \beta_j^{(s)} \) in the LQA
objective function, leading to

$$\beta^{(g+1)} = \arg\max \left\{ \sum_{i=1}^{n} (y_i - x_i^T \beta)^2 - \frac{1}{2} \sum_{j=1}^{p} g'(|\beta_j|) |\beta_j| \right\}.$$ 

By analogy with the above proposition, it is straightforward to demonstrate that LLA is an exact EM algorithm whenever the penalty function is a scale-mixture of double-exponentials, such as for the NEG penalty described by Griffin and Brown [2005] and Scheipl and Kneib [2009].

### 3.3 ECME for jointly estimating $\beta$ and $\tau$

The expectation–conditional maximization algorithm [Meng and Rubin, 1993] is a generalization of EM that can be used when there are multiple sets of parameters to be estimated. The ECM algorithm replaces the $M$-step with a sequence of conditional maximization (CM) steps that each maximize $Q$ with respect to one set of parameters, conditional on the current values of the others. Liu and Rubin [1994] showed that the ECM algorithm converges faster if conditional maximizations of $Q$ are replaced by conditional maximizations of the observed data posterior. Liu and Rubin called this the ECME algorithm, with the final “E” referring to conditional maximization of either function.

An ECME algorithm for learning $\beta$ and $\tau$ together can be obtained by assuming a prior distribution $p(\nu)$, where $\nu$ is a transformation $g_*(\tau)$ corresponding to the penalty function $g(\beta_j/\tau s_j)$. This transformation will satisfy the equivalence

$$g(\beta_j/\tau s_j) = g(\beta_j/s_j)/g_*(\tau)$$

for all values of $\tau$. For example, if $g(\beta_j/\tau s_j) = |\beta_j/\tau s_j|^a$ is the bridge penalty function, then $\nu = g_*(\tau) = \tau^a$, since

$$\left| \frac{\beta_j}{\tau s_j} \right|^a = \frac{|\beta_j/s_j|^a}{\nu}.$$ 

This provides an alternative to the approach for handling $\nu$ in Candes and Tao [2007] and Belloni and Chernozhukhov [2010], who recommend pre-specified fixed choices that relate strongly to the universal threshold.

Since $p(\beta_j \mid \tau)$ is a location-scale family, we have

$$p(\beta_j \mid \tau) \propto \frac{1}{\tau} \exp\{-g(\beta_j/\tau s_j)\} = \nu^{-1/a} \exp\{-g(\beta_j/s_j)/\nu\},$$

up to constants not involving $\beta_j$ or $\tau$. If the penalty function is the bridge/ℓa penalty, for example, then a conjugate inverse-gamma IG$(a_\nu, b_\nu)$ prior can be placed upon $\nu$, leading to the conditional posterior

$$p(\nu \mid \beta) \sim IG \left( a_\nu + p/a, b_\nu + \sum_{j=1}^{p} g(\beta_j/s_j) \right)$$

directly from (2), which has a closed-form conditional mean.

Under this prior one may estimate $\nu$ with a minor modification of the previous algorithm, summarized in the figure above. The CME step could be replaced by a CM step that estimates $\nu$ in terms of the latent variables $\lambda_j^{-1}$, but this would delay convergence.
3.4 Speeding up the EM/ECME convergence

Our primary goal in this paper is to show the relevance of the conditionally Gaussian representation of (3)–(4), together with Theorem 2.2, for fitting a wide class of regularized estimators within a unified variance–mean mixture framework. We have therefore focused only on the most basic implementations of the above computational methods.

There are many further variants on the basic EM and Gibbs algorithms, some of which can lead to dramatic speed advantages while maintaining stability, which we have not explored here. Ghosh and Dunson [2009], for example, show how some of these ideas can be exploited in sampling the variance components in Bayesian factor models, which pose many of the same issues with convergence that arise here. A key reference is Meng and van Dyk [1997]. Other algorithms include MDA [Liu, 1995], PXDA [Liu and Wu, 1999], MM [Hunter and Lange, 2000], the partially collapsed Gibbs sampler [van Dyk and Park, 2008], and the MCMC alternatives to EM discussed by van Dyk and Meng [2011]. Many of these modifications depend upon additional analytical work for particular choices of g and f—for example, the marginalizations carried out by Gelman et al. [2005] for the robit model.

In cases where \( p > n \), it is possible avoid inverting a \( p \times p \) matrix at each step of the algorithm by recognizing that

\[
\left( \tau^{-2}S^{-1}\hat{\Lambda}^{-1}(g) + X^T\hat{\Omega}^{-1}(g)X \right)^{-1} = \tau^2S\Lambda \left\{ I_n - \tau^2X^T(\Omega + XS\Lambda X^T)^{-1}XS\Lambda \right\},
\]

in accordance with the Sherman–Morrison–Woodbury identity. In large problems this will speed convergence considerably.

3.5 The posterior mean

In many situations—for example, estimation of \( \beta \) under squared-error loss—the relevant quantity of interest is the posterior mean, not the mode. For example, both Hans [2009] and Efron [2009] raise the point that, for the purposes of predicting future observables, the posterior mean can be the best choice for estimating \( \beta \).

The following theorem represents the posterior mean for \( \beta \) in terms of the score function of the predictive distribution, generalizing the results of Brown [1971], Masreliez [1975], and Pericchi and Smith [1992], and Carvalho et al. [2010]. There are a number of possible versions of such a theorem. Here we consider a variance–mean mixture prior \( p(\beta_j) \) with a general location likelihood \( p(y - \beta) \), but clearly a similar result holds the other way around. We consider the case where \( X \) is an orthogonal matrix, assumed without loss of generality to be the identity matrix (in which case we apply the univariate theorem component by component). The generalization to nonorthogonal designs is straightforward, following the original Masreliez [1975] paper; see, for example, Griffin and Brown [2010].

**Theorem 3.3.** Let \( p(|y - \beta_j|) \) be the likelihood for a location parameter \( \beta_j \), symmetric in \( y - \beta \), and let \( p(\beta_j) = \int \phi(\beta_j; \mu_\beta + \kappa_\beta\lambda_j, \tau^2s^2_j\lambda_j) \ p(\lambda_j) \ d\lambda_j \) be a normal variance–mean mixture prior. Define the
following distributions:

\[
m(y) = \int p(y - \beta_j) p(\beta_j) \, d\beta_j
\]

\[
p^*(\lambda_j) = \frac{\lambda_j^{-1} p(\lambda_j)}{E(\lambda_j^{-1})}
\]

\[
p^*(\beta_j) = \int \phi(\beta_j; \mu + \kappa \lambda_j, \lambda_j) p^*(\lambda_j)
\]

\[
m^*(y) = \int p(y - \beta_j) p^*(\beta_j).
\]

Then

\[
E(\beta_j | y) = -\frac{\kappa \beta_j \tau^2 s_j^2}{\tau^2 s_j^2} + \frac{\mu \beta_j E(\lambda_j^{-1})}{\tau^2 s_j^2} m^*(y) + \frac{E(\lambda_j^{-1}) m^*(y)}{\tau^2 s_j^2} \frac{\partial \log m^*(y)}{\partial y}.
\]

(18)

4 A conjugate-gradient variation

The computational bottleneck of Algorithm 1 arises from the need to repeatedly solve the linear system \( A^{(g)} \beta^{(g+1)} = b^{(g)} \) for \( \beta^{(g+1)} \), where \( A \) and \( b \) change at every step. This operation is \( O(d^3) \), where \( d = \min\{n, p\} \). Moreover, it must be repeated \( T \) times, where \( T \) is the number of global iterations of the ECM algorithm needed to converge.

Because \( A^{(g)} \) and \( b^{(g)} \) change at every step, however, it is inefficient to solve the system for \( \beta \) exactly. An approximate solution will still be sufficient for global convergence, as long as this approximate solution improves the observed-data objective function compared to the previous iteration. This will ensure that the sequence \( \{Q(\beta^{(1)}), Q(\beta^{(2)}), \ldots\} \) is monotonically decreasing.

Such an approximate solution can be found using the conjugate-gradient algorithm; see, for example Golub and Van Loan [1996]. We outline this modification in Figure 2. The conjugate-gradient algorithm’s computational bottleneck involves a series matrix-vector multiplications, each of which is \( O(d^2) \). This offers two major advantages.

1. If the algorithm is allowed to proceed for exactly \( d \) steps, then an exact solution (subject to floating-point error) will be produced in \( O(d^3) \) total operations. But typically far fewer than \( d \) steps are necessary to reach a good approximate solution.

2. In most major software languages, there are open-source libraries that implement parallel matrix-vector multiplication. This fact allows users to exploit a multi-core processing environment without having to manage concurrent data-access issues.

5 Examples

5.1 Bridge estimation with heavy-tailed errors

We first consider an example involving a bridge/\( \ell^a \) penalty, assuming heavy-tailed errors. As a test data set, we used the ozone data available from the R package faraway. The \( y \) variable is the maximum daily ozone concentration observed in Los Angeles for 330 days in 1976, while
Algorithm 2: Tilted, iteratively re-weighted conjugate gradient

Given a starting value $\beta^{(1)}$:

For iteration $g = 1, 2, \ldots$

Update $\Omega^{-1}$ as in Algorithm 1.

Update $\Lambda^{-1}$ as in Algorithm 1.

If $\lambda_j^{-1} > \lambda_{\text{max}}^{-1}$, delete $\lambda_j$ and $\beta_j$ from the model.

Update $\beta$:

First set

$$A^{(g)} := \tau^{-2} \Lambda^{-1(g)} + X^T \Omega^{-1(g)} X$$

$$b^{(g)} := X^T \left( \Omega^{-1(g)} y - \mu \omega^{-1}(g) \right)$$

for regression, or

$$A^{(g)} := \tau^{-2} \Lambda^{-1(g)} + X^T \Omega^{-1(g)} X_*$$

$$b^{(g)} := X_*^T \Omega^{-1(g)} \left( \mu \omega + \kappa \hat{\omega}(g) \right)$$

for classification. Then initialize the following inner-loop terms:

$$\beta^{(g,0)} := \beta^{(g-1)}$$

$$r^{(g,0)} := b - A^{(g)} \beta^{(g,0)}$$

$$d^{(g,0)} := r^{(g)}$$

$$c^{(g,0)} := A^{(g)} d^{(g,0)}$$

$$\alpha^{(g,0)} := \frac{r^{(g,0)} T r^{(g,0)}}{d^{(g,0)} T c^{(g,0)}}$$

$$\Delta^{(g,0)} := \alpha^{(g,0)} d^{(g,0)}$$

While $|\Delta^{(g,J)}| > \delta_{\text{min}}$, increment $l$ and set

$$\beta^{(g,J)} := \beta^{(g,J-1)} + \Delta^{(g,J-1)}$$

$$r^{(g,J)} := r^{(g,J-1)} - \alpha^{(g,J-1)} c^{(g,J-1)}$$

$$\gamma^{(g,J)} := \frac{r^{(g,J)} T r^{(g,J)}}{r^{(g,J-1)} T r^{(g,J-1)}}$$

$$d^{(g,J)} := r^{(g,J)} + \gamma^{(g,J)} d^{(g,J-1)}$$

$$c^{(g,J)} := A^{(g)} d^{(g,J)}$$

$$\alpha^{(g,J)} := \frac{r^{(g,J)} T r^{(g,J)}}{d^{(g,J)} T c^{(g,J)}}$$

$$\Delta^{(g,J)} := \alpha^{(g,J)} d^{(g,J)}$$

Set $\beta^{(g)} = \beta^{(g,J)}$.

End when the sequence of parameter estimates $\{\beta^{(1)}, \beta^{(2)}, \ldots\}$ has converged.

Figure 2: Tilted, iteratively re-weighted conjugate gradient for fitting sparse non-Gaussian models.
Figure 3: Solution path for the traditional lasso estimator (top) and the bridge estimator with heavy-tailed errors (bottom).
the predictors are 8 atmospheric quantities and their squared values (hence $p = 16$). To fit a regularized linear model to this data, we minimize the objective function

$$
Q(\beta) = \sum_{i=1}^{n} \left( \frac{d + 1}{2} \right) \log \left( 1 + \frac{(y_i - x_i^T \beta)^2}{d \sigma^2} \right) + \sum_{j=1}^{p} \left| \frac{\beta_j}{\tau} \right|^a,
$$

corresponding to a Student-$t$ error model with $d$ degrees of freedom and an $\ell^\alpha$ constraint on the $\beta$ sequence. We investigate the behavior of the solution for $\alpha = 1/2$ over a range of $\tau$ values, with $\sigma$ and $\beta$ estimated by their joint mode for every value of $\tau$. As a benchmark, we consider the traditional lasso estimator.

The EM algorithm described in Section 3 is straightforward to implement for this estimator. Applying Theorem 3.2 leads to conditional moments of the form

$$\hat{\omega}_i^{-1} = \frac{d + 1}{d \sigma^2 + z_i^2},$$

$$\hat{\lambda}_j^{-1} = \alpha (\tau s_j)^{2-a} |\beta_j|^{a-2}.$$

This second moment follows from the form of the bridge penalty,

$$p(\beta_j \mid \tau) = \exp \left( - |\beta_j / \tau|^a \right)$$

$$\frac{\partial}{\partial \beta_j} \log p(\beta_j \mid \tau) = - \frac{\alpha}{\tau^a} |\beta_j|^{a-1} \text{sign}(\beta_j),$$

where $\tau s_j$ replaces $\tau$ in the presence of known scale factors.
Table 2: Average sum of squared errors in estimating $\beta$ on 100 simulated data sets for each of two signal classes.

<table>
<thead>
<tr>
<th>Signal Type</th>
<th>MLE</th>
<th>Lasso</th>
<th>gDP</th>
</tr>
</thead>
<tbody>
<tr>
<td>r-spike signal</td>
<td>21.00</td>
<td>7.04</td>
<td>3.96</td>
</tr>
<tr>
<td>linear-decay signal</td>
<td>110.8</td>
<td>40.8</td>
<td>32.4</td>
</tr>
</tbody>
</table>

Figure 3 shows the solution path for both the lasso and bridge estimators as a function of the maximum $\ell^1$ norm of the coefficient vector. The primary differences concern the relative sizes of variables 1, 2, and 4 over the solution path. Part of these differences can be attributed to the relative tail weight of the implied likelihood under each model. Figure 4 shows how the estimated $\omega_i$’s under the Student-t/bridge model naturally account for heteroscedasticity in the data.

5.2 Classification with a generalized double-Pareto penalty

We consider two approaches to regularized classification: support vector machines, and logistic regression.

As the previous results demonstrate, the support vector-machine objective function has a conditionally Gaussian representation, where $\omega_i$ has an improper generalized inverse-Gaussian prior with $(\alpha, \kappa, \mu) = (1, 1, 1)$. This leads to

$$-(1/2) \log p(z_i) = (1/2)|1 - z_i| + (1/2)(1 - z_i) = \max(1 - z_i, 0),$$

recalling that $z_i = y_i x_i^T \beta$. Taking derivatives, for $z \neq 0$ we get

$$(1 - y_i x_i^T \beta) \mathbb{E}(\omega_i^{-1}|\beta(s)) = \pm 1$$

by applying Theorem 3.2, leading to

$$\hat{\omega}_i^{-1} = |1 - y_i x_i^T \beta|^{-1},$$

an expression which leads to a trivial EM implementation for regularized SVM’s. This offers an alternative to the Bayesian approach for SVM’s described in Mallick et al. [2005].

For the logistic criterion function, recall that $\omega$ has a Polya distribution with $\alpha = 1, \kappa = 1/2$. Theorem 3.2 gives the relevant conditional moment as

$$\hat{\omega}_i^{-1} = \frac{1}{z_i} \left\{ \frac{e^{z_i}}{1 + e^{z_i}} - \frac{1}{2} \right\},$$

which has a limit of $1/4$ at the origin.

As a penalty function, we use the generalized double-Pareto model proposed by Armagan et al. [2010], where

$$p(\beta_j | \tau) \propto \left( 1 + \frac{|\beta_j|}{\alpha \tau} \right)^{-(1+\alpha)}.$$

This prior has a spike at zero like the Laplace density, but also has a Student-t-like tail behavior. Armagan et al. [2010] show that this model leads to strong consistency of the posterior in
regression models with a diverging number of parameters, which can be thought of as a Bayesian analogue of the oracle property.

The generalized double-Pareto model has a conditionally Gaussian representation, making Theorem 3.2 applicable, and leading to

$$\hat{\lambda}^{-1}_j = \frac{1 + \alpha}{\alpha \tau |\beta_j| + \beta_j^2}.$$ 

This conditional moment can be incorporated easily into an EM mode-finder for any variance-mean mixture error model.

To give an illustration of the method, we compared the double-Pareto penalty to the well-known lasso penalty on two kinds of simulated sparse data sets. The first corresponded to a so-called “r-spike” signal with $p = 25, r = 5, n = 125$, where the first 5 entries of $\beta$ were equal to $\sqrt{p/r}$ and the next 20 were identically zero. This ensures that $\|\beta\|^2 = p$. The second case was a “linear signal decay” problem, where $\beta = (10, 9, \ldots, 2, 1, 0, 0, \ldots 0)^T$, with $p = 50$ and $n = 200$.

In each case, we simulated 100 different data sets, where in each case the design matrix was filled in with standard normal random variables. We fit a penalized logistic regression model, with $\tau$ chosen by ordinary cross-validation. The results of these simulations are described in Table 2, which shows the average sum of squared errors in estimating the sparse set of regression coefficients. In each case the gDP penalty appears to beat the lasso penalty despite being no harder to fit here, suggesting that there may be room for studying alternative penalty functions in classification problems.

5.3 Penalized quantile regression

To generate the pseudo-likelihood corresponding to quantile regression, choose $p(\omega_i)$ to be a generalized inverse-Gaussian prior where $(\alpha, x, \mu) = (1, 1 - 2q, 0)$. This leads to

$$-\log p(z_i) = |z_i| + (2q - 1)z_i,$$

which in turn leads to quantile regression for the $q$th quantile [Koenker 2005]. [Li et al. 2010] provide a Bayesian approach for estimating a quantile regression function by explicitly dealing with the check-loss likelihood. In contrast we represent the check-loss function as a mixture of normals, with a generalized inverse-Gaussian mixing distribution. This can be combined with any penalty function on $\beta$ within the class.

Applying Theorem 3.2 we get

$$\hat{\omega}_i^{-1} = |y_i - x_i^T \beta^{(g)}|^{-1}$$

as the conditional EM update. Meanwhile, the conditional $\beta$ estimate is

$$\beta^{(g+1)} = \left(\tau^{-2} S^{-1} \hat{A}^{-1} + X^T \hat{\Omega}^{-1} X\right)^{-1} (X^T \hat{\Omega}^{-1} Y - (1 - 2q) 1),$$

where the tilting of the observations depends upon the desired quantile $q$.

To illustrate the method, we simulated 50 data sets with $p = 25, n = 50$, and $\beta = (5, 4, 3, 2, 1, 0, \ldots, 0)^T$. In each case the 90th percentile of the data was a linear function of $\beta$ with i.i.d. $\sim \mathcal{N}(0, 1)$ design.
points. Noise was added by simulating errors from a normal distribution whose 90th percentile was the linear predictor $x^T \beta$, and whose variance was $\sigma^2 = 5^2$. For each data set, we fit three models: the maximum likelihood quantile regression (using the R package quantreg); quantile regression with a lasso penalty; and quantile regression with a generalized double-Pareto penalty with $\alpha = 3$. For the second and third method, the scale of regularization $\tau$ was chosen by cross validation. We measured performance by squared-error loss in estimating $\beta$, and out-of-sample check loss on a new data set with the same value of $\beta$ but different design points and residuals.

The results are in Table 3. Both regularized versions of quantile regression for the 90th percentile seem to outperform the straight estimator. No significant difference emerged between the lasso and gDP penalties in terms of performance, although the gDP solutions were systematically more sparse.

6 Bayesian approaches

The quantity $p(\beta) = e^{-Q(\beta)}$ can also be interpreted as a posterior distribution in a well-specified Bayesian inference problem. Under this interpretation, the posterior mean estimator $\hat{\beta}_{\text{Bayes}} = E_\pi(\beta \mid y)$ is also of interest, particularly in terms of prediction [Park and Casella, 2008; Efron, 2009; Gramacy and Polson, 2010; Hans, 2009; Polson and Scott, 2011a].

Our TIRLS algorithm also extends easily to MCMC, which must be tailored to specific distributional forms of $p(\omega)$ and $p(\lambda)$. A preliminary lemma establishes that this class is closed under Bayesian learning, and aids in constructing efficient Gibbs-sampling algorithms for estimating $\beta$ based on simple mixtures of ridge regressions.

Lemma 6.1. Suppose that the objective function $Q(\beta)$ can be represented by a hierarchical variance–mean Gaussian mixture, as in Equations (3) and (4). Then the pseudo-posterior distribution $p(\beta \mid \tau, y)$ is also variance–mean mixture:

$$p(\beta \mid \tau, y) = \int_{\mathbb{R}_{p+n}^+} \phi \left( \beta \mid b_{(\omega, \lambda)}, B_{(\omega, \lambda)} \right) p(\omega, \lambda \mid \tau, y) d\omega d\lambda.$$  

This follows trivially from the fact that $p(\beta \mid \omega, \lambda, y)$ is conditionally Gaussian, a fact easily verified using Equations (10) and (11).

In building MCMC algorithms tailored to draw from

$$p(\lambda \mid \beta, \tau) = \frac{p(\lambda, \beta \mid \tau)}{p(\beta \mid \tau)},$$  

Algorithm: MCMC-RR

Step 1: Draw $\beta^{(g+1)} \sim p(\beta \mid \tau, \Lambda^{(g)}, \lambda^{(g)}, y) \sim N\left(b^{(g)}, B^{(g)}\right)$.

For regression,

$$
B^{(g)} = \left(\tau^{-2}S^{-1}\hat{\Lambda}^{-1(\hat{g})} + X^T\hat{\Omega}^{-1(\hat{g})}X\right)^{-1}
$$

$$
b^{(g)} = \left(\tau^{-2}S^{-1}\hat{\Lambda}^{-1(\hat{g})} + X^T\hat{\Omega}^{-1(\hat{g})}X\right)^{-1}X^T\hat{\Omega}^{-1(\hat{g})} \left(y - \mu_z1 - \kappa\hat{\omega}\right)
$$

For classification,

$$
B^{(g)} = \left(\tau^{-2}S^{-1}\hat{\Lambda}^{-1(\hat{g})} + X^T\hat{\Omega}^{-1(\hat{g})}X\right)^{-1}
$$

$$
b^{(g)} = \left(\tau^{-2}S^{-1}\hat{\Lambda}^{-1(\hat{g})} + X^T\hat{\Omega}^{-1(\hat{g})}X\right)^{-1}\mu_z1 + \kappa\hat{\omega}\right)
$$

Step 2: Draw $\omega^{(g+1)} \sim p(\omega \mid \beta^{(g+1)}, y)$ for $1 \leq i \leq n$, from its full conditional distribution.

Step 3: Draw $\lambda^{(g+1)} \sim p(\lambda \mid \beta^{(g+1)}, y)$ for $1 \leq i \leq p$, from its full conditional distribution. Alternatively, use the fact that the regularization penalty $p(\beta \mid \tau)$ is known in closed form to draw directly from

$$
p(\lambda \mid \beta, \tau) = \frac{p(\lambda, \beta \mid \tau)}{p(\beta \mid \tau)} .
$$

Step 4: Under a bridge penalty, draw $\nu$ from its full conditional distribution

$$
(v \mid \beta) \sim IG\left(a_v + p/a, b_v + \sum_{j=1}^{p} g(\beta_j/z_i)\right)
$$

under an inverse-gamma prior. Alternatively, simulate $p(\tau \mid \beta, \lambda)$ from its full conditional distribution.
for specific models \( p(\lambda_j) \), one may exploit the fact that \( p(\beta \mid \tau) \) is usually specified in closed form for a given regularized estimator. Therefore \( p(\lambda_j \mid \beta, \tau, y) \) is nearly in closed form:

\[
p(\lambda_j \mid \beta, \tau, y) = \frac{\phi(\beta_j \mid \mu_\beta + \kappa_\beta \lambda_j, \tau^2 \sqrt{\lambda_j}) p(\lambda_j \mid \tau)}{p(\beta_j \mid \tau)},
\]

where \( p(\beta_j \mid \tau) \) is specified by the regularization penalty and is known. The slice sampler [Neal, 2003] is an effective general way of simulating from distributions of this form. Given the conditional normality of \( \beta \) and \( z \), if we can simulate from the joint posterior \( p(\omega, \lambda \mid \tau, y) \), then we can simulate from the posterior distribution for \( \beta \).

Moreover, inspection of the complete-data log posterior distribution (12) shows that the conditional likelihood in \( \omega_j \) takes an inverse-Gaussian form. This will lead to closed-form posterior updates for a wide range of possible priors, including all generalized inverse-Gaussian distributions. When \( p(\omega_j) \) is a Polya distribution, or any other distribution for which the prior does not combine with the inverse-Gaussian likelihood, the method described in Algorithm 3.2 of [Godsill, 2000] can be used to draw from the full conditional of \( \omega_j \), assuming that it is possible to simulate from the prior distribution.

Finally, we can also draw from \( p(\nu \mid \beta) \), assuming an inverse gamma prior. If \( \nu \sim IG(a_\nu, b_\nu) \), then Equation (17) specifies the full conditionally conjugate draw. This draw can then be mapped to the equivalent scale parameter \( \tau \).

### 7 Discussion

In this paper, we have extended the class of regularized estimators based on normal variance mixtures in two ways: first, by generalizing to normal variance–mean mixtures, and second, by using a similar mixture representation for both the likelihood and the prior. The latent variables in the mixture representation result in simple EM and ECME algorithms that can provide point estimates of parameters, which has been our focus on this paper. But we have also pointed out where the same representation facilitates MCMC sampling regimes that can bring the theory of Bayesian linear models to bear on a broad class of error models and penalty functions.

We have also described several examples in regularized regression and classification that illustrate the generality of the framework proposed here, and in particular the broad utility of Theorem 3.2. Of particular interest is the ability this gives practitioners to combine, in a more-or-less arbitrary fashion, any pseudo-likelihood and pseudo-prior from the family of normal mean–variance mixtures, while still remaining with a simple conditionally Gaussian, weighted-least-squares framework. Our approach can be generalized to include spike-and-slab priors for regression coefficients, along the lines of [Polson and Scott, 2011b]. It also easily accommodates basis expansions using kernels in place of \( x_i \).

The approach can also be generalized to include mixture models, for which EM algorithms are well established. For example, one often encounters a mixture of multinomials, where the likelihood can be written as

\[
\prod_{n=1}^{N} \left\{ \sum_{i=1}^{K} \theta_i \prod_{j=1}^{V} \beta_{ij}^{w_{ijn}} \right\} = \prod_{n=1}^{N} \sum_{i=1}^{K} \theta_i e^{\sum_{j=1}^{V} w_{ijn} \ln \beta_{ij}},
\]
for mixture weights $\theta$, log probabilities $\beta_{ij}$, and indicators $w_{jn}$, as for a model of words within topics within documents. EM algorithms for such models are notoriously difficult [e.g. Ng and McLachlan 2004], and extending the methods described here to these domains is an active area of future research.

A Proofs

A.1 Theorem 3.1

We demonstrate the result for a regression problem, with the classification result following as a simple modification.

Write the complete-data log posterior distribution as

$$
\log p(\beta | \omega, \lambda, \tau, y) = c_0(\omega, \lambda, y, \tau) - \frac{1}{2} \sum_{i=1}^{n} \omega_i^{-1} (z_i - \mu_z - \kappa \omega_i)^2 \\
- \frac{1}{2\tau^2} \sum_{j=1}^{p} \lambda_j^{-1} (\beta_j - \mu_\beta - \kappa \lambda_j)^2
$$

for some constant $c_0$ and known scaling factors $s_j$. We can factorize this further (as a function of $\beta$) as

$$
\log p(\beta | \omega, \lambda, \tau, y) = c_1(\omega, \lambda, y, \tau) - \frac{1}{2} \sum_{i=1}^{n} \omega_i^{-1} (z_i - \mu_z)^2 + \kappa y \sum_{i=1}^{n} (z_i - \mu_z) \\
- \frac{1}{2\tau^2} \sum_{j=1}^{k} \lambda_j^{-1} (\beta_j - \mu_\beta)^2 + \kappa \beta \sum_{j=1}^{p} s_j^{-2} (\beta_j - \mu_\beta),
$$

which depends linearly upon $\omega_i^{-1}$ and $\lambda_j^{-1}$.

Collecting terms, we can represent the log posterior (up to an additive constant not involving $\beta$) as a sum of quadratic forms in $\beta$:

$$
\log p(\beta | \omega, \lambda, \tau, y) = -\frac{1}{2} \left( \{y - \mu_z 1 - \kappa \omega\} - X\beta \right)^T \Omega^{-1} \left( \{y - \mu_z 1 - \kappa \omega\} - X\beta \right) \\
- \frac{1}{2\tau^2} \left( \beta - \mu_\beta 1 - \kappa \lambda \right)^T \Lambda^{-1} \left( \beta - \mu_\beta 1 - \kappa \lambda \right).
$$

This is the log posterior under a normal prior $\beta \sim \mathcal{N}(\mu_\beta 1 + \kappa \lambda, \tau^2 \Lambda)$. By using the expressions given in, for example, Lindley and Smith [1972], along with the simple identity $\Omega^{-1}(\mu 1 + \kappa \omega) = \mu \omega + \kappa 1$, we arrive at the result.

For classification, on the other hand, let $X_*$ be the matrix with rows $x_i^* = y_i \cdot x_i$. The kernel of the conditionally normal likelihood then becomes

$$
(X_* \beta - \mu_z 1 - \kappa \omega)^T \Omega^{-1} (X_* \beta - \mu_z 1 - \kappa \omega)^T.
$$

Hence it is as if we observe the $n$-dimensional “data” vector $\mu_z 1 + \kappa \omega$ in a regression model having design matrix $X_*$, and we proceed by a similar argument to arrive at the result.
The rest of the argument follows the standard Masreliez approach.

We obtain the desired result by using the identities

\[
\frac{\partial \phi(\beta_j | \mu_{\beta} + \kappa_{\beta} \lambda_j, \tau^2 s_j^2 \lambda_j)}{\partial \beta_j} = -\beta_j - \mu_{\beta} - \kappa_{\beta} \lambda_j \phi(\beta_j | \mu_{\beta} + \kappa_{\beta} \lambda_j, \tau^2 s_j^2 \lambda_j).
\]

We use this fact to differentiate

\[
p(\beta_j | \tau) = \int_{0}^{\infty} \phi(\beta_j | \mu_{\beta} + \kappa_{\beta} \lambda_j, \tau^2 s_j^2 \lambda_j) p(\lambda_j | \tau) d\lambda_j
\]

under the integral sign:

\[
\frac{\partial}{\partial \beta_j} p(\beta_j | \tau) = \int_{0}^{\infty} \frac{\partial}{\partial \beta_j} \phi(\beta_j | \mu_{\beta} + \kappa_{\beta} \lambda_j, \tau^2 s_j^2 \lambda_j) p(\lambda_j | \tau) d\lambda_j.
\]

Dividing by \( p(\beta_j | \tau) \) and using the above identity for the inner function, we get

\[
\frac{\partial}{\partial \beta_j} p(\beta_j | \tau) = \frac{\kappa_{\beta}}{\tau^2 s_j^2} p(\beta_j | \tau) - \frac{\beta_j - \mu_{\beta}}{\tau^2 s_j^2} \mathbb{E} \left( \lambda_j^{-1} | \beta(\theta), \tau, y_i \right).
\]

Hence we can in general find the inverse moment needed for the \( E \)-step, which can be calculated using the expression

\[
\frac{1}{p(\beta_j | \tau)} \frac{\partial}{\partial \beta_j} p(\beta_j | \tau) = \frac{\kappa_{\beta}}{\tau^2 s_j^2} - \frac{\beta_j - \mu_{\beta}}{\tau^2 s_j^2} \mathbb{E} \left( \lambda_j^{-1} | \beta(\theta), \tau, y \right).
\]

Equivalently, in terms of the penalty function \( \log p(\beta_j | \tau) \), we have

\[
(\beta_j - \mu_{\beta}) \mathbb{E} \left( \lambda_j^{-1} | \beta_j \right) = \kappa_{\beta} - \tau^2 s_j^2 \frac{\partial}{\partial \beta_j} \log p(\beta_j | \tau).
\]

By a similar argument, we also have

\[
(z_i - \mu_{z_i}) \mathbb{E} \left( \omega_j^{-1} | \beta, z_i \right) = \kappa_{z_i} - \sigma^2 \frac{\partial}{\partial z_i} \log p(z_i | \beta),
\]

We obtain the desired result by using the identities

\[
\frac{\partial}{\partial z_i} \log p(z_i | \beta_i) = f'(\beta_i | \tau) \quad \text{and} \quad \frac{\partial}{\partial \beta_j} \log p(\beta_j | \tau) = g'(\beta_j | \tau)
\]

Our extension of Masreliez’s theorem to variance–mean mixtures follows a similar path. From before, since \( \phi \) is a normal kernel,

\[
\frac{\partial \phi(\beta_j | \mu_{\beta} + \kappa_{\beta} \lambda_j, \tau^2 s_j^2 \lambda_j)}{\partial \beta_j} = -\beta_j - \mu_{\beta} - \kappa_{\beta} \lambda_j \phi(\beta_j | \mu_{\beta} + \kappa_{\beta} \lambda_j, \tau^2 s_j^2 \lambda_j).
\]

Differentiating under the integral sign and applying this result, we have that

\[
\frac{1}{\tau^2 s_j^2 \lambda_j} \beta_j \phi(\beta_j | \mu_{\beta} + \kappa_{\beta} \lambda_j, \tau^2 s_j^2 \lambda_j) = \frac{\mu_{\beta} - \kappa_{\beta}}{\tau^2 s_j^2 \lambda_j} - \frac{\partial \phi(\beta_j | \mu_{\beta} + \kappa_{\beta} \lambda_j, \tau^2 s_j^2 \lambda_j)}{\partial \beta_j}.
\]

The rest of the argument follows the standard Masreliez approach.
B Distributional results

B.1 GIG/Hyperbolic distributions

The density of a generalized inverse-Gaussian random variable is
\[ p_{\text{GIG}}(v | \psi, \gamma, \delta) = C(\psi, \gamma, \delta) v^{\psi-1} \exp \left\{ -\frac{1}{2} \left( \frac{\delta^2}{v} + \gamma^2 v \right) \right\}, \]
where \( C(\psi, \gamma, \delta) \) is a normalization constant not depending on \( v \). The class of generalized inverse-Gaussian distributions has the inverse-Gaussian as a special case, when \( \psi = -1/2 \). It also nests all gamma and inverse-gamma distributions.

Under this mixing measure, \( p(\theta) \) is a generalized hyperbolic distribution, denoted \( \mathcal{GH}(\mu, \kappa, \gamma, \delta, \psi) \). Its density function is
\[ p(\theta | \mu, \kappa, \gamma, \delta, \psi) = C(\psi, \gamma, \delta) \frac{K_{1/2}}{\{\{\delta^2 + (\theta - \mu)^2/\alpha\}^{1/2}\}^{1/2}} \cdot \exp\{\kappa(\theta - \mu)\}, \]
where \( \alpha^2 = \gamma^2 + \kappa^2 \) and \( K_{1/2} \) is a modified Bessel function of the second kind. Recall that \( \mu \) and \( \kappa \) are the mean and drift parameters of the normal variance–mean mixture, respectively.

B.2 Polya/Z distributions

The underlying density is \( p_{\text{PY}}(v | \alpha, \alpha - 2 \kappa) = \sum_{k=0}^\infty w_k e^{-a_k v} \). The terms in this sum are
\[ a_k = \frac{(\alpha + k)(\kappa + k)}{2} \quad \text{and} \quad w_k = d_k \prod_{j \neq k} \left( \frac{a_k}{a_j - a_k} \right) = \left( \frac{-2\delta}{k} \right) \frac{(\delta + k)}{B(\delta + b, \delta - b)}, \]
where \( b = \frac{1}{2}(\alpha - \kappa), \delta = \frac{1}{2}(\alpha + \kappa) \), and \( \binom{-2\delta}{k} = \frac{(-1)^k(2\delta)!(2\delta + k - 1)!}{k!} \).

References


