Robust Probabilistic Modeling with Bayesian Data Reweighting

Yixin Wang [1] **Alp Kucukelbir** [1] **David M. Blei** [1]

Abstract

Probabilistic models analyze data by relying on a set of assumptions. Data that exhibit deviations from these assumptions can undermine inference and prediction quality. Robust models offer protection against mismatch between a model's assumptions and reality. We propose a way to systematically detect and mitigate mismatch of a large class of probabilistic models. The idea is to raise the likelihood of each observation to a weight and then to infer both the latent variables and the weights from data. Inferring the weights allows a model to identify observations that match its assumptions and down-weight others. This enables robust inference and improves predictive accuracy. We study four different forms of mismatch with reality, ranging from missing latent groups to structure misspecification. A Poisson factorization analysis of the Movielens 1M dataset shows the benefits of this approach in a practical scenario.

1. Introduction

Probabilistic modeling is a powerful approach to discovering hidden patterns in data. We begin by expressing assumptions about the class of patterns we expect to discover; this is how we design a probability model. We follow by inferring the posterior of the model; this is how we discover the specific patterns manifest in an observed data set. Advances in automated inference (Hoffman & Gelman, 2014; Mansinghka et al., 2014; Kucukelbir et al., 2017) enable easy development of new models for machine learning and artificial intelligence (Ghahramani, 2015).

In this paper, we present a recipe to robustify probabilistic models. What do we mean by "robustify"? Departure from a model's assumptions can undermine its inference and prediction performance. This can arise due to corrupted observations, or in general, measurements that do not belong to the process we are modeling. Robust models should perform well in spite of such mismatch with reality.

Consider a movie recommendation system. We gather data of people watching movies via the account they use to log in. Imagine a situation where a few observations are corrupted For example, a child logs in to her account and regularly watches popular animated films. One day, her parents use the same account to watch a horror movie. Recommendation models, like Poisson factorization (PF), struggle with this kind of corrupted data (see Section 4): it begins to recommend horror movies.

What can be done to detect and mitigate this effect? One strategy is to design new models that are less sensitive to corrupted data, such as by replacing a Gaussian likelihood with a heavier-tailed t distribution (Huber, 2011; Insua & Ruggeri, 2012). Most probabilistic models we use have more sophisticated structures; these template solutions for specific distributions are not readily applicable. Other classical robust techniques act mostly on distances between observations (Huber, 1973); these approaches struggle with high-dimensional data. How can we still make use of our favorite probabilistic models while making them less sensitive to the messy nature of reality?

Main idea. We propose reweighted probabilistic models (RPM). The idea is simple. First, posit a probabilistic model. Then adjust the contribution of each observation by raising each likelihood term to its own (latent) weight. Finally, infer these weights along with the latent variables of the original probability model. The posterior of this adjusted model identifies observations that match its assumptions; it downweights observations that disagree with its assumptions.

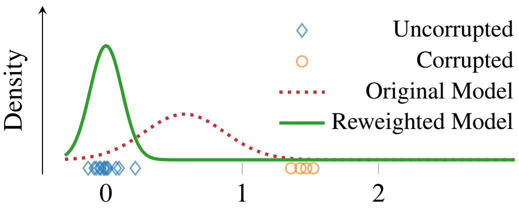


Figure 1. Fitting a unimodal distribution to a dataset with corrupted measurements. The RPM downweights the corrupted observations.

Figure 1 depicts this tradeoff. The dataset includes cor-

[1] Columbia University, New York City, USA. Correspondence to: Yixin Wang <yixin.wang@columbia.edu>.

Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017. Copyright 2017 by the author(s).

rupted measurements that undermine the original model; Bayesian data reweighting automatically trades off the low likelihood of the corrupted data near 1.5 to focus on the uncorrupted data near zero. The RPM (green curve) detects this mismatch and mitigates its effect compared to the poor fit of the original model (red curve).

Formally, consider a dataset of N independent observations $y = (y_1, \dots, y_N)$. The likelihood factorizes as a product $\prod_{n=1}^{N} \ell(y_n \mid \beta)$, where β is a set of latent variables. Posit a prior distribution $p_\beta(\beta)$.

Bayesian data reweighting follows three steps:

1. Define a probabilistic model $p_\beta(\beta) \prod_{n=1}^{N} \ell(y_n \mid \beta)$.
2. Raise each likelihood to a positive latent weight w_n. Then choose a prior on the weights $p_w(w)$, where $w = (w_1, \dots, w_N)$. This gives a reweighted probabilistic model (RPM)

$$p(y, \beta, w) = \frac{1}{Z} p_\beta(\beta) p_w(w) \prod_{n=1}^{N} \ell(y_n \mid \beta)^{w_n},$$

   where Z is the normalizing factor.
3. Infer the posterior of both the latent variables β and the weights w, $p(\beta, w \mid y)$.

The latent weights w allow an RPM to automatically explore which observations match its assumptions and which do not. Writing out the logarithm of the RPM gives some intuition; it is equal (up to an additive constant) to

$$\log p_\beta(\beta) + \log p_w(w) + \sum_n w_n \log \ell(y_n \mid \beta). \tag{1}$$

Posterior inference, loosely speaking, seeks to maximize the above with respect to β and w. The prior on the weights $p_w(w)$ plays a critical role: it trades off extremely low likelihood terms, caused by corrupted measurements, while encouraging the weights to be close to one. We study three options for this prior in Section 2.

How does Bayesian data reweighting induce robustness? First, consider how the weights w affect Equation (1). The logarithm of our priors are dominated by the $\log w_n$ term: this is the price of moving w_n from one towards zero. By shrinking w_n, we gain an increase in $w_n \log \ell(y_n \mid \beta)$ while paying a price in $a \log w_n$. The gain outweighs the price we pay if $\log \ell(y_n \mid \beta)$ is very negative. Our priors are set to prefer w_n to stay close to one; an RPM only shrinks w_n for very unlikely (e.g., corrupted) measurements.

Now consider how the latent variables β affect Equation (1). As the weights of unlikely measurements shrink, the likelihood term can afford to assign low mass to those corrupted measurements and focus on the rest of the dataset. Jointly, the weights and latent variables work together to automatically identify unlikely measurements and focus on observations that match the original model's assumptions.

Section 2 presents these intuitions in full detail, along with theoretical corroboration. In Section 3, we study four models under various forms of mismatch with reality, including missing modeling assumptions, misspecified nonlinearities, and skewed data. RPMs provide better parameter inference and improved predictive accuracy across these models. Section 4 presents a recommendation system example, where we improve on predictive performance and identify atypical film enthusiasts in the Movielens 1M dataset.

Related work. Jerzy Neyman elegantly motivates the main idea behind robust probabilistic modeling, a field that has attracted much research attention in the past century.

> Every attempt to use mathematics to study some real phenomena must begin with building a mathematical model of these phenomena. Of necessity, the model simplifies matters to a greater or lesser extent and a number of details are ignored. [...] The solution of the mathematical problem may be correct and yet it may be in violent conflict with realities simply because the original assumptions of the mathematical model diverge essentially from the conditions of the practical problem considered. (Neyman, 1949, p.22).

Our work draws on three themes around robust modeling.

The first is a body of work on robust statistics and machine learning (Provost & Fawcett, 2001; Song et al., 2002; Yu et al., 2012; McWilliams et al., 2014; Feng et al., 2014; Shafieezadeh-Abadeh et al., 2015). These developments focus on making specific models more robust to imprecise measurements.

One strategy is popular: localization. To localize a probabilistic model, allow each likelihood to depend on its own "copy" of the latent variable β_n. This transforms the model into

$$p(y, \beta, \alpha) = p_\alpha(\alpha) \prod_{n=1}^{N} \ell(y_n \mid \beta_n) p_\beta(\beta_n \mid \alpha), \tag{2}$$

where a top-level latent variable α ties together all the β_n variables (de Finetti, 1961; Wang & Blei, 2015).[1] Localization decreases the effect of imprecise measurements. RPMs present a broader approach to mitigating mismatch, with improved performance over localization (Sections 3 and 4).

The second theme is robust Bayesian analysis, which studies sensitivity with respect to the prior (Berger et al., 1994).

[1] Localization also relates to James-Stein shrinkage; Efron (2010) connects these dots.

Recent advances directly focus on sensitivity of the posterior (Minsker et al., 2014; Miller & Dunson, 2015) or the posterior predictive distribution (Kucukelbir & Blei, 2015). We draw connections to these ideas throughout this paper.

The third theme is data reweighting. This involves designing individual reweighting schemes for specific tasks and models. Consider robust methods that toss away "outliers." This strategy involves manually assigning binary weights to datapoints (Huber, 2011). Another example is covariate shift adaptation/importance sampling where reweighting transforms data to match another target distribution (Veach & Guibas, 1995; Sugiyama et al., 2007; Shimodaira, 2000; Wen et al., 2014). In contrast, RPMs treat weights as latent variables. The weights are automatically inferred; no custom design is required. RPMs also connect to ideas around ensemble learning and boosting (Schapire & Freund, 2012). Boosting procedures reweight datapoints to build an ensemble of predictors for supervised learning, whereas RPMs apply to Bayesian models in general.

2. Reweighted Probabilistic Models

Reweighted probabilistic models (RPM) offer a new approach to robust modeling. The idea is to automatically identify observations that match the assumptions of the model and to base posterior inference on these observations.

2.1. Definitions

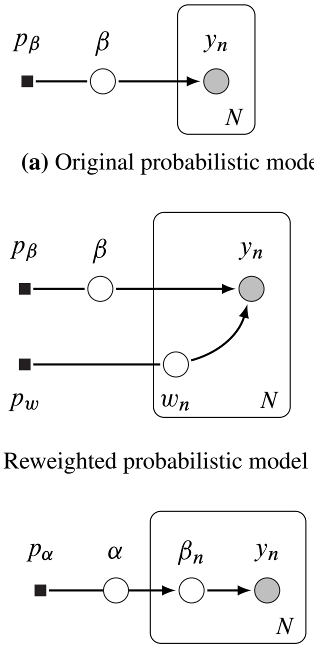
An RPM scaffolds over a probabilistic model, $p_\beta(\beta)\prod_{n=1}^N \ell(y_n \mid \beta)$. Raise each likelihood to a latent weight and posit a prior on the weights. This gives the reweighted joint density

$$p(y, \beta, w) = \frac{1}{Z} p_\beta(\beta) p_w(w) \prod_{n=1}^{N} \ell(y_n \mid \beta)^{w_n}, \qquad (3)$$

where $Z = \int p_\beta(\beta) p_w(w) \prod_{n=1}^N \ell(y_n \mid \beta)^{w_n} \,\mathrm{d}y \,\mathrm{d}\beta \,\mathrm{d}w$ is the normalizing factor.

The reweighted density integrates to one when the normalizing factor Z is finite. This is always true when the likelihood $\ell(\cdot \mid \beta)$ is an exponential family distribution with Lesbegue base measure (Bernardo & Smith, 2009); this is the class of models we study in this paper.[2]

RPMs apply to likelihoods that factorize over the observations. (We discuss non-exchangeable models in Section 5.) Figure 2 depicts an RPM as a graphical model. Specific models may have additional structure, such as a separation of local and global latent variables (Hoffman et al., 2013), or fixed parameters; we omit these in this figure.



(a) Original probabilistic model

(b) Reweighted probabilistic model (RPM)

(c) Localized probabilistic model

Figure 2. RPMs begin with a probabilistic model **(a)** and introduce a set of weights w as latent variables. This gives a model **(b)** that explores which data observations match its assumptions. Localization **(c)**, instead, builds a hierarchical model. (Appendix A shows when a localized model is also an RPM.)

The reweighted model introduces a set of weights; these are latent variables, each with support $w_n \in \mathbb{R}_{>0}$. To gain intuition, consider how these weights affect the posterior, which is proportional to the product of the likelihood of every measurement. A weight w_n that is close to zero flattens out its corresponding likelihood $\ell(y_n \mid \beta)^{w_n}$; a weight that is larger than one makes its likelihood more peaked. This, in turn, enables the posterior to focus on some measurements more than others. The prior $p_w(w)$ ensures that not too many likelihood terms get flattened; in this sense, it plays an important regularization role.

We study three options for this prior on weights: a bank of Beta distributions, a scaled Dirichlet distribution, and a bank of Gamma distributions.

Bank of Beta priors. This option constrains each weight as $w_n \in (0, 1)$. We posit an independent prior for each weight

$$p_w(w) = \prod_{n=1}^{N} \mathrm{Beta}(w_n \,;\, a, b) \qquad (4)$$

and use the same parameters a and b for all weights. This is the most conservative option for the RPM; it ensures that none of the likelihoods ever becomes more peaked than it was in the original model.

[2] Heavy-tailed likelihoods and Bayesian nonparametric priors may violate this condition; we leave these for future analysis.

The parameters a, b offer an expressive language to describe different attitudes towards the weights. For example, setting both parameters less than one makes the Beta act like a "two spikes and a slab" prior, encouraging weights to be close to zero or one, but not in between. As another example, setting a greater than b encourages weights to lean towards one.

Scaled Dirichlet prior. This option ensures the sum of the weights equals N. We posit a symmetric Dirichlet prior on all the weights

$$\begin{aligned} w &= Nv \\ p_v(v) &= \text{Dirichlet}(a\mathbf{1}) \end{aligned} \tag{5}$$

where a is a scalar parameter and $\mathbf{1}$ is a $(N \times 1)$ vector of ones. In the original model, where all the weights are one, then the sum of the weights is N. The Dirichlet option maintains this balance; while certain likelihoods may become more peaked, others will flatten to compensate.

The concentration parameter a gives an intuitive way to configure the Dirichlet. Small values for a allow the model to easily up- or down-weight many data observations; larger values for a prefer a smoother distribution of weights. The Dirichlet option connects to the bootstrap approaches in Rubin et al. (1981); Kucukelbir & Blei (2015), which also preserves the sum of weights as N.

Bank of Gamma priors. Here we posit an independent Gamma prior for each weight

$$p_w(w) = \prod_{n=1}^{N} \text{Gamma}(w_n \; ; \, a, b) \tag{6}$$

and use the same parameters a and b for all weights. We do not recommend this option, because observations can be arbitrarily up- or down-weighted. In this paper, we only consider Equation (6) for our theoretical analysis in Section 2.2.

The bank of Beta and Dirichlet options perform similarly. We prefer the Beta option as it is more conservative, yet find the Dirichlet to be less sensitive to its parameters. We explore these options in the empirical study (Section 3).

2.2. Theory and intuition

How can theory justify Bayesian data reweighting? Here we investigate its robustness properties. These analyses intend to confirm our intuition from Section 1. Appendices B and C present proofs in full technical detail.

Intuition. Recall the logarithm of the RPM joint density from Equation (1).Now compute the maximum-a-posterior (MAP) estimate of the weights w. The partial derivative is

$$\frac{\partial \log p(y, \beta, w)}{\partial w_n} = \frac{\mathrm{d} \log p_w(w_n)}{\mathrm{d} w_n} + \log \ell(y_n \mid \beta) \tag{7}$$

for all $n = 1, \ldots, N$. Plug the Gamma prior from Equation (6) into the partial derivative in Equation (7) and set it equal to zero. This gives the MAP estimate of w_n,

$$\widehat{w}_n = \frac{a - 1}{b - \log \ell(y_n \mid \beta)}. \tag{8}$$

The MAP estimate $\widehat{w}_n$ is an increasing function of the log likelihood of y_n when $a > 1$. This reveals that $\widehat{w}_n$ shrinks the contribution of observations that are unlikely under the log likelihood; in turn, this encourages the MAP estimate for $\widehat{\beta}$ to describe the majority of the observations. This is how an RPM makes a probabilistic model more robust.

A similar argument holds for other exponential family priors on w with $\log w_n$ as a sufficient statistic. We formalize this intuition and generalize it in the following theorem, which establishes sufficient conditions where a RPM improves the inference of its latent variables β.

Theorem 1 *Denote the true value of β as β^*. Let the posterior mean of β under the weighted and unweighted model be $\bar{\beta}_w$ and $\bar{\beta}_u$ respectively. Assume mild conditions on p_w, ℓ and the corruption level, and that $|\ell(y_n \mid \bar{\beta}_w) - \ell(y_n \mid \beta^*)| < \epsilon$ holds $\forall n$ with high probability. Then, there exists an N^* such that for $N > N^*$, we have $|\bar{\beta}_u - \beta^*| \succeq_2 |\bar{\beta}_w - \beta^*|$, where $\succeq_2$ denotes second order stochastic dominance. (Details in Appendix B.)*

The likelihood bounding assumption is common in robust statistics theory; it is satisfied for both likely and unlikely (corrupted) measurements. How much of an improvement does it give? We can quantify this through the influence function (IF) of $\bar{\beta}_w$.

Consider a distribution G and a statistic $T(G)$ to be a function of data that comes iid from G. Take a fixed distribution, e.g., the population distribution, F. Then, $\text{IF}(z; T, F)$ measures how much an additional observation at z affects the statistic $T(F)$. Define

$$\text{IF}(z; T, F) = \lim_{t \to 0^+} \frac{T(t\delta_z + (1-t)F) - T(F)}{t}$$

for z where this limit exists. Roughly, the IF measures the asymptotic bias on $T(F)$ caused by a specific observation z that does not come from F. We consider a statistic T to be robust if its IF is a bounded function of z, i.e., if outliers can only exert a limited influence (Huber, 2011).

Here, we study the IF of the posterior mean $T = \bar{\beta}_w$ under the true data generating distribution $F = \ell(\cdot \mid \beta^*)$. Say a value z has likelihood $\ell(z \mid \beta^*)$ that is nearly zero; we think of this z as corrupted. Now consider the weight function induced by the prior $p_w(w)$. Rewrite it as a function of the log likelihood, like $w(\log \ell(\cdot \mid \beta^*))$ as in Equation (8).

Theorem 2 *If $\lim_{a \to -\infty} w(a) = 0$ and $\lim_{a \to -\infty} a \cdot w(a) < \infty$, then $\text{IF}(z; \bar{\beta}_w, \ell(\cdot \mid \beta^*)) \to 0$ as $\ell(z \mid \beta^*) \to 0$.*

This result shows that an RPM is robust in that its IF goes to zero for unlikely measurements. This is true for all three priors. (Details in Appendix C.)

2.3. Inference and computation

We now turn to inferring the posterior of an RPM, $p(\beta, w \mid y)$. The posterior lacks an analytic closed-form expression for all but the simplest of models; even if the original model admits such a posterior for β, the reweighted posterior may take a different form.

To approximate the posterior, we appeal to probabilistic programming. A probabilistic programming system enables a user to write a probability model as a computer program and then compile that program into an inference executable. Automated inference is the backbone of such systems: it takes in a probability model, expressed as a program, and outputs an efficient algorithm for inference. We use automated inference in Stan, a probabilistic programming system (Carpenter et al., 2015).

In the empirical study that follows, we highlight how RPMs detect and mitigate various forms of model mismatch. As a common metric, we compare the predictive accuracy on held out data for the original, localized, and reweighted model.

The posterior predictive likelihood of a new datapoint $y_\dagger$ is $p_{\text{original}}(y_\dagger \mid y) = \int \ell(y_\dagger \mid \beta) p(\beta \mid y)\,\mathrm{d}\beta$. Localization couples each observation with its own copy of the latent variable; this gives $p_{\text{localized}}(y_\dagger \mid y) = \iint \ell(y_\dagger \mid \beta_\dagger) p(\beta_\dagger \mid \alpha) p(\alpha \mid y)\,\mathrm{d}\alpha\,\mathrm{d}\beta_\dagger$ where $\beta_\dagger$ is the localized latent variable for the new datapoint. The prior $p(\beta_\dagger \mid \alpha)$ has the same form as p_β in Equation (2).

Bayesian data reweighting gives the following posterior predictive likelihood

$$p_{\text{RPM}}(y_\dagger \mid y) = \iint p(y_\dagger \mid \beta, w_\dagger) p_{\text{RPM}}(\beta \mid y) p(w_\dagger)\ \mathrm{d}w_\dagger\,\mathrm{d}\beta,$$

where $p_{\text{RPM}}(\beta \mid y)$ is the marginal posterior, integrating out the inferred weights of the training dataset, and the prior $p(w_\dagger)$ has the same form as p_w in Equation (3).

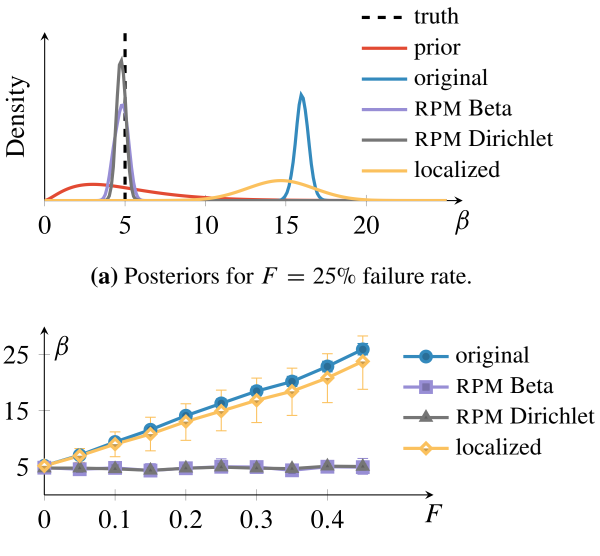
3. Empirical Study

We study RPMs under four types of mismatch with reality. This section involves simulations of realistic scenarios; the next section presents a recommendation system example using real data. We default to No-U-Turn sampler (NUTS) (Hoffman & Gelman, 2014) for inference in all experiments, except for Sections 3.5 and 4 where we leverage variational inference (Kucukelbir et al., 2017). The additional computational cost of inferring the weights is unnoticeable relative to inference in the original model.

3.1. Outliers: a network wait-time example

A router receives packets over a network and measures the time it waits for each packet. Suppose we typically observe wait-times that follow a Poisson distribution with rate $\beta = 5$. We model each measurement using a Poisson likelihood $\ell(y_n \mid \beta) = \text{Poisson}(\beta)$ and posit a Gamma prior on the rate $p_\beta(\beta) = \text{Gam}(a = 2, b = 0.5)$.

Imagine that $F\%$ percent of the time, the network fails. During these failures, the wait-times come from a Poisson with much higher rate $\beta = 50$. Thus, the data actually contains a mixture of two Poisson distributions; yet, our model only assumes one. (Details in Appendix D.1.)



(a) Posteriors for $F = 25\%$ failure rate.

(b) Posterior 95% credible intervals.

Figure 3. Outliers simulation study. We compare Beta(0.1, 0.01) and Dir(**1**) as priors for the reweighted probabilistic model. **(a)** Posterior distributions on β show a marked difference in detecting the correct wait-time rate of $\beta = 5$. **(b)** Posterior 95% confidence intervals across failure rates F show consistent behavior for both Beta and Dirichlet priors. ($N = 100$ with 50 replications.)

How do we expect an RPM to behave in this situation? Suppose the network failed 25% of the time. Figure 3a shows the posterior distribution on the rate β. The original posterior is centered at 18; this is troubling, not only because the rate is wrong but also because of how confident the posterior fit is. Localization introduces greater uncertainty, yet still estimates a rate around 15. The RPM correctly identifies that the majority of the observations come from $\beta = 5$. Observations from when the network failed are down-weighted. It gives a confident posterior centered at five.

Figure 3b shows posterior 95% credible intervals of β under failure rates up to $F = 45\%$. The RPM is robust to corrupted measurements; instead it focuses on data that it can explain within its assumptions. When there is no corruption, the RPM performs just as well as the original model.

Visualizing the weights elucidates this point. Figure 4 shows the posterior mean estimates of w for $F = 25\%$. The weights are sorted into two groups, for ease of viewing. The weights of the corrupted observations are essentially zero; this downweighting is what allows the RPM to shift its posterior on β towards five.

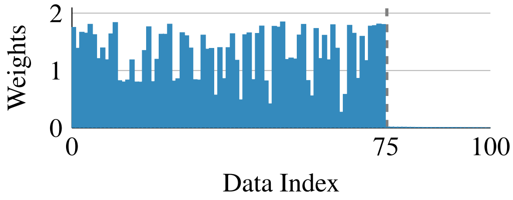


Figure 4. Posterior means of the weights w under the Dirichlet prior. For visualization purposes, we sorted the data into two groups: the first 75 contain observations from the normal network; the remaining 25 are the observations when the network fails.

Despite this downweighting, the RPM posteriors on β are not overdispersed, as in the localized case. This is due to the interplay we described in the introduction. Downweighting observations should lead to a smaller effective sample size, which would increase posterior uncertainty. But the downweighted datapoints are corrupted observations; including them also increases posterior uncertainty.

The RPM is insensitive to the prior on the weights; both Beta and Dirichlet options perform similarly. From here on, we focus on the Beta option. We let the shape parameter a scale with the data size N such that $N/a \approx 10^3$; this encodes a mild attitude towards unit weights. We now move on to other forms of mismatch with reality.

3.2. Missing latent groups: predicting color blindness

Color blindness is unevenly hereditary: it is much higher for men than for women (Boron & Boulpaep, 2012). Suppose we are not aware of this fact. We have a dataset of both genders with each individual's color blindness status and his/her relevant family history. No gender information is available. Consider analyzing this data using logistic regression. It can only capture one hereditary group. Thus, logistic regression misrepresents both groups, even though men exhibit strong heredity. In contrast, an RPM can *detect* and *mitigate* the missing group effect by focusing on the dominant hereditary trait. Here we consider men as the dominant group.

We simulate this scenario by drawing binary indicators of color blindness $y_n \sim \text{Bernoulli}(1/1 + \exp(-p_n))$ where the p_n's come from two latent groups: men exhibit a stronger dependency on family history ($p_n = 0.5x_n$) than women ($p_n = 0.01x_n$). We simulate family history as $x_n \sim \text{Unif}(-10, 10)$. Consider a Bayesian logistic regression model without intercept. Posit a prior on the slope as $p_\beta(\beta) = \mathcal{N}(0, 10)$ and assume a Beta(0.1, 0.01) prior on the weights. (Details in Appendix D.2.)

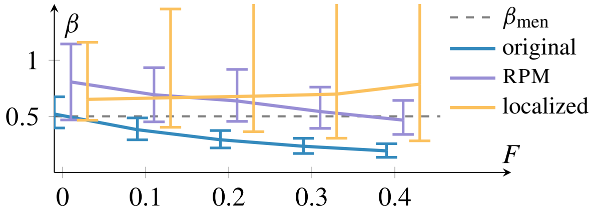


Figure 5. Missing latent groups study. Posterior 95% credible intervals for the RPM always include the dominant $\beta_{\text{men}} = 0.5$, as we vary the percentage of females in the data. Dataset size $N = 100$ with 50 replications.

Figure 5 shows the posterior 95% credible intervals of β as we vary the percentage of females from $F = 0\%$ to 40%. A horizontal line indicates the correct slope for the dominant group, $\beta_{\text{men}} = 0.5$. As the size of the missing latent group (women) increases, the original model quickly shifts its credible interval away from 0.5. The reweighted and localized posteriors both contain $\beta_{\text{men}} = 0.5$ for all percentages, but the localized model exhibits much higher variance in its estimates.

This analysis shows how RPMs can mitigate the effect of missing latent groups. While the original logistic regression model would perform equally poorly on both groups, an RPM is able to automatically focus on the dominant group.

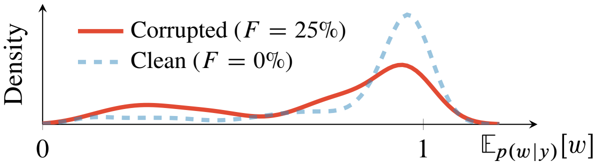


Figure 6. Kernel density estimate of the distribution of weights across all measurements in the missing latent groups study. The percentage of females is denoted by F. A hypothetical clean dataset receives weights that concentrate around one; the actual corrupted dataset exhibits a two-hump distribution of weights.

An RPM also functions as a diagnostic tool to *detect* mismatch with reality. The distribution of the inferred weights indicates the presence of datapoints that defy the assumptions of the original model. Figure 6 shows a kernel density estimate of the inferred posterior weights. A hypothetical dataset with no corrupted measurements receives weights close to one. In contrast, the actual dataset with measurements from a missing latent group exhibit a bimodal distribution of weights. Testing for bimodality of the inferred weights is one way in which an RPM can be used to diagnose mismatch with reality.

True structure	Model structure	Original mean(std)	RPM mean(std)	Localization mean(std)
$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2$	$\beta_0 + \beta_1 x_1 + \beta_2 x_2$	3.16(1.37)	**2.20**(1.25)	2.63(1.85)
$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_2^2$	$\beta_0 + \beta_1 x_1 + \beta_2 x_2$	30.79(2.60)	**16.32**(1.96)	21.08(5.20)
$\beta_0 + \beta_1 x_1 + \beta_2 x_2$	$\beta_0 + \beta_1 x_1$	**0.58**(0.38)	0.60(0.40)	0.98(0.54)

Table 1. RPMs improve absolute deviations of posterior mean β_1 estimates. (50 replications.)

3.3. Covariate dependence misspecification: a lung cancer risk study

Consider a study of lung cancer risk. While tobacco usage exhibits a clear connection, other factors may also contribute. For instance, obesity and tobacco usage appear to interact, with evidence towards a quadratic dependence on obesity (Odegaard et al., 2010).

Denote tobacco usage as x_1 and obesity as x_2. We study three models of lung cancer risk dependency on these covariates. We are primarily interested in understanding the effect of tobacco usage; thus we focus on β_1, the regression coefficient for tobacco. In each model, some form of covariance misspecification discriminates the true structure from the assumed structure.

For each model, we simulate a dataset of size $N = 100$ with random covariates $x_1 \sim \mathcal{N}(10, 5^2)$ and $x_2 \sim \mathcal{N}(0, 10^2)$ and regression coefficients $\beta_{0,1,2,3} \sim \text{Unif}(-10, 10)$. Consider a Bayesian linear regression model with prior $p_\beta(\beta) = \mathcal{N}(0, 10)$. (Details in Appendix D.3.)

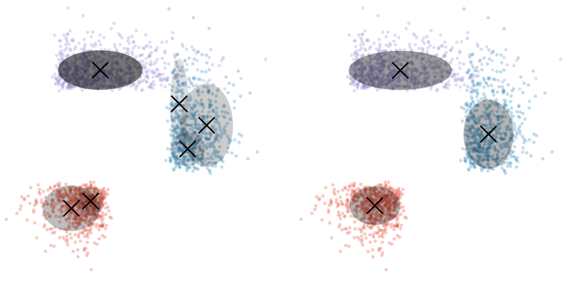
Table 1 summarizes the misspecification and shows absolute differences on the estimated β_1 regression coefficient. The RPM yields better estimates of β_1 in the first two models. These highlight how the RPM leverages datapoints useful for estimating β_1. The third model is particularly challenging because obesity is ignored in the misspecified model. Here, the RPM gives similar results to the original model; this highlights that RPMs can only use available information. Since the original model lacks dependence on x_2, the RPM cannot compensate for this.

3.4. Predictive likelihood results

Table 2 shows how RPMs also improve predictive accuracy. In all the above examples, we simulate test data with and without their respective types of corruption. RPMs improve prediction for both clean and corrupted data, as they focus on data that match the assumptions of the original model.

3.5. Skewed data: cluster selection in a mixture model

Finally, we show how RPMs handle skewed data. The Dirichlet process mixture model (DPMM) is a versatile model for density estimation and clustering (Bishop, 2006; Murphy, 2012). While real data may indeed come from a finite mixture of clusters, there is no reason to assume each cluster is distributed as a Gaussian. Inspired by the experiments in Miller & Dunson (2015), we show how a reweighted DPMM reliably recovers the correct number of components in a mixture of skewnormals dataset.



(a) Original model **(b)** RPM

Figure 7. A finite approximation DPMM to skewnormal distributed data that come from three groups. The shade of each cluster indicates the inferred mixture proportions ($N = 2000$).

A standard Gaussian mixture model (GMM) with large K and a sparse Dirichlet prior on the mixture proportions is an approximation to a DPMM (Ishwaran & James, 2012). We simulate three clusters from two-dimensional skewnormal distributions and fit a GMM with maximum $K = 30$. Here we use automatic differentiation variational inference (ADVI), as NUTS struggles with inference of mixture models (Kucukelbir et al., 2017). (Details in Appendix D.4.)

Figure 7 shows posterior mean estimates from the original GMM; it incorrectly finds six clusters. In contrast, the RPM identifies the correct three clusters. Datapoints in the tails of each cluster get down-weighted; these are datapoints that do not match the Gaussianity assumption of the model.

4. Case Study: Poisson factorization for recommendation

We now turn to a study of real data: a recommendation system. Consider a video streaming service; data comes as a binary matrix of users and the movies they choose to watch. How can we identify patterns from such data? Poisson factorization (PF) offers a flexible solution (Cemgil, 2009; Gopalan et al., 2015). The idea is to infer a K-dimensional

	Outliers		Missing latent groups		Misspecified structure	
	Clean	Corrupted	Clean	Corrupted	Clean	Corrupted
Original model	−744.2	−1244.5	−108.6	−103.9	−136.3	−161.7
Localized model	−730.8	−1258.4	−53.6	−112.7	−192.5	−193.1
RPM	**−328.5**	**−1146.9**	**−43.9**	**−90.5**	**−124.1**	**−144.1**

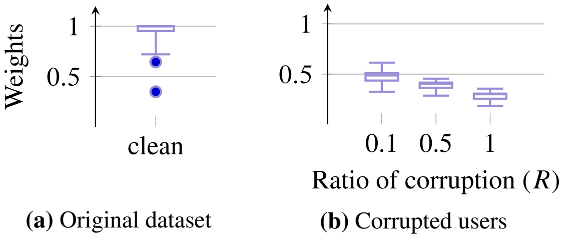
Table 2. Posterior predictive likelihoods of clean and corrupted test data. Outliers and missing latent groups have $F = 25\%$. The misspecified structure is missing the interaction term. Results are similar for other levels and types of mismatch with reality.

Average log likelihood	Corrupted users		
	0%	1%	2%
Original model	−1.68	−1.73	−1.74
RPM	**−1.53**	**−1.53**	**−1.52**

Table 3. Held-out predictive accuracy under varying amounts of corruption. Held-out users chosen randomly (20% of total users).

latent space of user preferences $\boldsymbol{\theta}$ and movie attributes $\boldsymbol{\beta}$. The inner product $\boldsymbol{\theta}^\top\boldsymbol{\beta}$ determines the rate of a Poisson likelihood for each binary measurement; Gamma priors on $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$ promote sparse patterns. As a result, PF finds interpretable groupings of movies, often clustered according to popularity or genre. (Full model in Appendix E.)

How does classical PF compare to its reweighted counterpart? As input, we use the MovieLens 1M dataset, which contains one million movie ratings from 6 000 users on 4 000 movies. We place iid Gamma$(1, 0.001)$ priors on the preferences and attributes. Here, we have the option of reweighting users or items. We focus on users and place a Beta$(100, 1)$ prior on their weights. For this model, we use MAP estimation. (Localization is computationally challenging for PF; it requires a separate "copy" of $\boldsymbol{\theta}$ for each movie, along with a separate $\boldsymbol{\beta}$ for each user. This dramatically increases computational cost.)



(a) Original dataset **(b)** Corrupted users

Figure 8. Inferred weights for clean and corrupted data. **(a)** Most users receive weights very close to one. **(b)** Corrupted users receive weights much smaller than one. Larger ratios of corruption R imply lower weights.

We begin by analyzing the original (clean) dataset. Reweighting improves the average held-out log likelihood from -1.68 of the original model to -1.53 of the corresponding RPM. The boxplot in Figure 8a shows the inferred weights. The majority of users receive weight one, but a few users are down-weighted. These are film enthusiasts who appear to indiscriminately watch many movies from many genres. (Appendix F shows an example.) These users do not contribute towards identifying movies that go together; this explains why the RPM down-weights them.

Recall the example from our introduction. A child typically watches popular animated films, but her parents occasionally use her account to watch horror films. We simulate this by corrupting a small percentage of users. We replace a ratio $R = (0.1, 0.5, 1)$ of these users' movies with randomly selected movies.

The boxplot in Figure 8b shows the weights we infer for these corrupted users, based on how many of their movies we randomly replace. The weights decrease as we corrupt more movies. Table 3 shows how this leads to higher held-out predictive accuracy; down-weighting these corrupted users leads to better prediction.

5. Discussion

Reweighted probabilistic models (RPM) offer a systematic approach to mitigating various forms of mismatch with reality. The idea is to raise each data likelihood to a weight and to infer the weights along with the hidden patterns. We demonstrate how this strategy introduces robustness and improves prediction accuracy across four types of mismatch.

RPMs also offer a way to *detect* mismatch with reality. The distribution of the inferred weights sheds light onto datapoints that fail to match the original model's assumptions. RPMs can thus lead to new model development and deeper insights about our data.

RPMs can also work with non-exchangeable data, such as time series. Some time series models admit exchangeable likelihood approximations (Guinness & Stein, 2013). For other models, a non-overlapping windowing approach would also work. The idea of reweighting could also extend to structured likelihoods, such as Hawkes process models.

Acknowledgements

We thank Adji Dieng, Yuanjun Gao, Inchi Hu, Christian Naesseth, Rajesh Ranganath, Francisco Ruiz, and Dustin Tran for their insightful comments. This work is supported by NSF IIS-1247664, ONR N00014-11-1-0651, DARPA PPAML FA8750-14-2-0009, DARPA SIMPLEX N66001-15-C-4032, and the Alfred P. Sloan Foundation.

References

Berger, James O, Moreno, Elías, Pericchi, Luis Raul, Bayarri, M Jesús, Bernardo, José M, Cano, Juan A, De la Horra, Julián, Martín, Jacinto, Ríos-Insúa, David, Betrò, Bruno, et al. An overview of robust Bayesian analysis. *Test*, 3(1):5–124, 1994.

Bernardo, José M and Smith, Adrian FM. *Bayesian Theory*. John Wiley & Sons, 2009.

Bishop, Christopher M. *Pattern Recognition and Machine Learning*. Springer New York, 2006.

Boron, Walter F and Boulpaep, Emile L. *Medical Physiology*. Elsevier, 2012.

Carpenter, Bob, Gelman, Andrew, Hoffman, Matt, Lee, Daniel, Goodrich, Ben, Betancourt, Michael, Brubaker, Marcus A, Guo, Jiqiang, Li, Peter, and Riddell, Allen. Stan: a probabilistic programming language. *Journal of Statistical Software*, 2015.

Cemgil, Ali Taylan. Bayesian inference for nonnegative matrix factorisation models. *Computational Intelligence and Neuroscience*, 2009.

de Finetti, Bruno. The Bayesian approach to the rejection of outliers. In *Proceedings of the Fourth Berkeley Symposium on Probability and Statistics*, 1961.

Efron, Bradley. *Large-Scale Inference*. Cambridge University Press, 2010.

Feng, Jiashi, Xu, Huan, Mannor, Shie, and Yan, Shuicheng. Robust logistic regression and classification. In *NIPS*. 2014.

Ghahramani, Zoubin. Probabilistic machine learning and artificial intelligence. *Nature*, 521(7553):452–459, 2015.

Gopalan, Prem, Hofman, Jake M, and Blei, David M. Scalable recommendation with hierarchical Poisson factorization. *UAI*, 2015.

Guinness, Joseph and Stein, Michael L. Transformation to approximate independence for locally stationary Gaussian processes. *Journal of Time Series Analysis*, 34(5): 574–590, 2013.

Hoffman, Matthew D and Gelman, Andrew. The No-U-Turn sampler. *Journal of Machine Learning Research*, 15(1): 1593–1623, 2014.

Hoffman, Matthew D, Blei, David M, Wang, Chong, and Paisley, John. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

Huber, Peter J. Robust regression: asymptotics, conjectures and Monte Carlo. *The Annals of Statistics*, pp. 799–821, 1973.

Huber, Peter J. *Robust Statistics*. Springer, 2011.

Insua, David Ríos and Ruggeri, Fabrizio. *Robust Bayesian Analysis*. Springer Science & Business Media, 2012.

Ishwaran, Hemant and James, Lancelot F. Approximate Dirichlet process computing in finite normal mixtures. *Journal of Computational and Graphical Statistics*, 2012.

Kucukelbir, Alp and Blei, David M. Population empirical Bayes. In *UAI*, 2015.

Kucukelbir, Alp, Tran, Dustin, Ranganath, Rajesh, Gelman, Andrew, and Blei, David M. Automatic differentiation variational inference. *Journal of Machine Learning Research*, 18(14):1–45, 2017.

Mansinghka, Vikash, Selsam, Daniel, and Perov, Yura. Venture: a higher-order probabilistic programming platform with programmable inference. *arXiv:1404.0099*, 2014.

McWilliams, Brian, Krummenacher, Gabriel, Lucic, Mario, and Buhmann, Joachim M. Fast and robust least squares estimation in corrupted linear models. In *NIPS*. 2014.

Miller, Jeffrey W and Dunson, David B. Robust Bayesian inference via coarsening. *arXiv preprint arXiv:1506.06101*, 2015.

Minsker, Stanislav, Srivastava, Sanvesh, Lin, Lizhen, and Dunson, David B. Robust and scalable Bayes via a median of subset posterior measures. *arXiv preprint arXiv:1403.2660*, 2014.

Murphy, Kevin P. *Machine Learning: a Probabilistic Perspective*. MIT Press, 2012.

Neyman, Jerzy. On the problem of estimating the number of schools of fish. In J. Neyman, M. Loeve and Yerushalmy, J. (eds.), *University of California Publications in Statistics*, volume 1, chapter 3, pp. 21–36. University of California Press, 1949.

Odegaard, Andrew O, Pereira, Mark A, Koh, Woon-Puay, Gross, Myron D, Duval, Sue, Mimi, C Yu, and Yuan, Jian-Min. BMI, all-cause and cause-specific mortality in

Chinese Singaporean men and women. *PLoS One*, 5(11), 2010.

Provost, Foster and Fawcett, Tom. Robust classification for imprecise environments. *Machine Learning*, 42(3): 203–231, 2001.

Rubin, Donald B et al. The Bayesian bootstrap. *The annals of statistics*, 9(1):130–134, 1981.

Schapire, R.E. and Freund, Y. *Boosting: Foundations and Algorithms*. MIT Press, 2012.

Shafieezadeh-Abadeh, Soroosh, Esfahani, Peyman Mohajerin, and Kuhn, Daniel. Distributionally robust logistic regression. In *NIPS*. 2015.

Shimodaira, Hidetoshi. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2): 227–244, 2000.

Song, Qing, Hu, Wenjie, and Xie, Wenfang. Robust support vector machine with bullet hole image classification. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 32(4):440–448, 2002.

Sugiyama, Masashi, Krauledat, Matthias, and Müller, Klaus-Robert. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8:985–1005, 2007.

Veach, Eric and Guibas, Leonidas J. Optimally combining sampling techniques for Monte Carlo rendering. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pp. 419–428. ACM, 1995.

Wang, Chong and Blei, David M. A general method for robust Bayesian modeling. *arXiv preprint arXiv:1510.05078*, 2015.

Wen, Junfeng, Yu, Chun-nam, and Greiner, Russell. Robust learning under uncertain test distributions: Relating covariate shift to model misspecification. In *ICML*, 2014.

Yu, Yaoliang, Aslan, Özlem, and Schuurmans, Dale. A polynomial-time form of robust regression. In *NIPS*. 2012.