Journal of Machine Learning Research 25 (2024) 1-35 Submitted 4/23; Revised 2/24; Published 3/24

Data Thinning for Convolution-Closed Distributions

Anna Neufeld ANEUFELD@FREDHUTCH.ORG
Public Health Sciences Division
Fred Hutchinson Cancer Center
Seattle, WA 98109, USA

Ameer Dharamshi ADHARAMS@UW.EDU
Department of Biostatistics
University of Washington
Seattle, WA 98195, USA

Lucy L. Gao LUCY.GAO@UBC.CA
Department of Statistics
University of British Columbia
Vancouver, British Columbia V6T 1Z4, Canada

Daniela Witten DWITTEN@UW.EDU
Departments of Statistics and Biostatistics
University of Washington
Seattle, WA 98195, USA

Editor: Jie Peng

Abstract

We propose data thinning, an approach for splitting an observation into two or more independent parts that sum to the original observation, and that follow the same distribution as the original observation, up to a (known) scaling of a parameter. This very general proposal is applicable to any convolution-closed distribution, a class that includes the Gaussian, Poisson, negative binomial, gamma, and binomial distributions, among others. Data thinning has a number of applications to model selection, evaluation, and inference. For instance, cross-validation via data thinning provides an attractive alternative to the usual approach of cross-validation via sample splitting, especially in settings in which the latter is not applicable. In simulations and in an application to single-cell RNA-sequencing data, we show that data thinning can be used to validate the results of unsupervised learning approaches, such as k-means clustering and principal components analysis, for which traditional sample splitting is unattractive or unavailable.

Keywords: Cross validation, sample splitting, clustering, principal component analysis

1. Introduction

As scientists fit increasingly complex models to their data, there is an ever-growing need for out-of-box methods that can be used to validate these models. In many settings, the most natural option is sample splitting, in which the n observations in a data set are split into a training set, used to fit a model, and a test set, used to validate it (Hastie et al., 2009). Sample splitting can also be applied to conduct inference after model selection (Rinaldo et al., 2019). Sample splitting is flexible and intuitive, and is a vital tool for any practicing data analyst.

However, in settings where there is one parameter of interest per observation, or the parameter of interest is a function of the n observations, sample splitting cannot be applied. For example, when estimating a low-rank approximation to a matrix, there is one parameter of interest (a latent

variable coordinate) for each of the n rows in the matrix. Similarly, in fixed-covariate regression under model misspecification, the target parameter depends on the specific n observations included in the data set (Buja et al., 2019). Finally, there may be settings in which we wish to draw observation-specific inferences about each of our n observations; sample splitting does not allow this.

In this paper, we consider an alternative to sample splitting that splits a single observation X into independent parts that follow the same distribution as X. Crucially, the fact that we split a *single* observation avoids the pitfalls of sample splitting in the situations mentioned above: we will split every observation in the data to obtain a training set and a test set that each involve all n observations.

The concept of splitting a single observation as an alternative to sample splitting has been explored in several recent papers (Rasines and Young, 2022; Leiner et al., 2023; Oliveira et al., 2021, 2022; Neufeld et al., 2022). We can split $X \sim \mathrm{N}(\mu, \sigma^2)$ with known σ^2 into two independent Gaussian random variables (Rasines and Young, 2022; Leiner et al., 2023; Oliveira et al., 2021), and $X \sim \text{Poisson}(\lambda)$ into two independent Poisson random variables (Neufeld et al., 2022; Leiner et al., 2023; Oliveira et al., 2022). However, outside of these two distributions, no proposals are available to split a random variable into independent parts that follow the same distribution as the original random variable. Leiner et al. (2023) propose data fission, a general-purpose approach to decompose X into two parts, $X^{(1)}$ and $X^{(2)}$, such that (i) $X^{(1)}$ and $X^{(2)}$ can together be used to reconstruct X, and (ii) the joint distribution of $\left(X^{(1)}, X^{(2)}\right)$ is tractable. However, outside of the special cases of the Gaussian and Poisson distributions, the proposal of Leiner et al. (2023) leads to $X^{(1)}$ and $X^{(2)}$ that (a) are not independent, and (b) whose distributions may or may not resemble that of X. As a consequence of both (a) and (b), unless the data are Gaussian- or Poisson-distributed, data fission does not serve as a direct alternative to sample splitting, as out-of-the-box tools can no longer be used for validation or inference. We elaborate on these points in Section 5.

In this paper, we propose data thinning, a recipe for decomposing a single observation X into two parts, $X^{(1)}$ and $X^{(2)}$, such that (i) $X = X^{(1)} + X^{(2)}$, (ii) $X^{(1)}$ and $X^{(2)}$ are independent, and (iii) $X^{(1)}$ and $X^{(2)}$ follow the same distribution as X, up to a (known) scaling of a parameter. Critically, properties (ii) and (iii) guarantee that this decomposition is straightforward to use in applied settings. For instance, to evaluate the suitability of a model for X, we can fit it to $X^{(1)}$ (since it follows the same distribution as X), and can validate it using $X^{(2)}$ (since it also follows the same distribution, and furthermore is independent of $X^{(1)}$). Our recipe can be applied to any distribution that is convolution-closed (Joe, 1996): this includes the multivariate Gaussian, Poisson, negative binomial, gamma, binomial, and multinomial distributions, among others. Thus, our work drastically expands the set of distributions that can be split into independent parts, and provides a unified lens through which to view seemingly unrelated approaches. Furthermore, data thinning can be used to decompose X into more than two independent random variables.

We illustrate our proposal with the following example, which shows that a gamma random variable can be thinned into M independent gamma random variables.

Example 1 (Gamma decomposition into M components, data thinning) *Suppose that* $X \sim \text{Gamma}(\alpha, \beta)$*, where* β *is unknown. We take* $(X^{(1)}, \ldots, X^{(M)}) = XZ$*, where* $Z \sim \text{Dirichlet}(\alpha/M, \ldots, \alpha/M)$*. Then* $X^{(1)}, \ldots, X^{(M)}$ *are mutually independent, they sum to* X*, and each is marginally drawn from a* $\text{Gamma}(\alpha/M, \beta)$ *distribution.*

In other words, data thinning allows us to decompose a $\text{Gamma}(\alpha, \beta)$ random variable, for which β is unknown, into M independent gamma random variables, $X^{(1)}, \ldots, X^{(M)}$. Therefore, fitting a model to $X - X^{(m)}$ and validating it using $X^{(m)}$ is straightforward.

The rest of this paper is organized as follows. In Section 2, we briefly review the class of convolution-closed distributions, and introduce a thinning procedure to split a single random variable into two independent random variables, each of which follows the same distribution as the original random variable (up to a scaling of the parameter(s)). We extend this thinning procedure to split a single random variable into an arbitrary number of independent random variables in Section 3. We elaborate on the comparison between sample splitting and data thinning in Section 4, and in Section 5 we elaborate on the comparison between data fission (Leiner et al., 2023) and data thinning. In Section 6, we focus on validating the results of clustering and low-rank matrix approximations. These are two settings in which the usual cross-validation via sample splitting approach cannot be directly applied (see, e.g. Owen and Perry, 2009; Fu and Perry, 2020), but data thinning provides a simple alternative. An application to single-cell RNA-sequencing data is in Section 7. We close with a discussion in Section 8. All proofs are in the appendix.

2. The Data Thinning Proposal

2.1 A Review of Convolution-Closed Distributions

We begin by defining a convolution-closed distribution (Joe, 1996; Jørgensen and Song, 1998).

Definition 1 (Convolution-closed) *Let F_λ denote a distribution indexed by a parameter λ in parameter space Λ. Let $X' \sim F_{\lambda_1}$ and $X'' \sim F_{\lambda_2}$ with $X' \perp\!\!\!\perp X''$. If $X' + X'' \sim F_{\lambda_1+\lambda_2}$ whenever $\lambda_1 + \lambda_2 \in \Lambda$, then F_λ is* convolution-closed *in the parameter λ.*

Many well-known distributions are convolution-closed. While the Poisson(λ) distribution is convolution-closed in its single parameter λ and the $\mathrm{N}(\mu, \sigma^2)$ distribution is convolution-closed in the two-dimensional parameter (μ, σ^2), other distributions, such as the gamma, are convolution-closed in just one parameter with the other parameter(s) held fixed. Table 1 provides details about some well-known convolution-closed distributions. The following definition provides a useful property of most convolution-closed distributions.

Definition 2 (Linear expectation property) *Let F_λ denote a distribution indexed by $\lambda \in \Lambda$. We say that it satisfies the linear expectation property if, for $X \sim F_\lambda$, $E[X]$ is a linear function of λ.*

Remark 3 (Most convolution-closed distributions satisfy the linear expectation property) *Let F_λ be a convolution-closed distribution whose first moment exists for all $\lambda \in \Lambda$. By definition, if $X' \sim F_{\lambda_1}$ and $X'' \sim F_{\lambda_2}$ and $\lambda_1 + \lambda_2 \in \Lambda$, then $X' + X'' \sim F_{\lambda_1+\lambda_2}$. By properties of the expected value, $E[X' + X''] = E[X'] + E[X'']$. Thus, the expectation is additive in λ, and so, outside of contrived counterexamples, F_λ satisfies the linear expectation property. The linear expectation property is satisfied for all distributions in Table 1.*

Remark 4 (Not all convolution-closed distributions satisfy the linear expectation property) *The* Cauchy(μ, γ) *distribution is convolution-closed in the two-dimensional parameter (μ, γ), but does not satisfy the linear expectation property because $E[X]$ does not exist.*

For a convolution-closed distribution F_λ, suppose that $X' \sim F_{\lambda_1}$ and $X'' \sim F_{\lambda_2}$ with $X' \perp\!\!\!\perp X''$. Let $G_{\lambda_1,\lambda_2,x}$ denote the conditional distribution of $X' \mid X' + X'' = x$. The density of the distribution $G_{\lambda_1,\lambda_2,x}$ can be written down for any F_λ with a known density function (Jørgensen, 1992). Furthermore, it turns out that $G_{\lambda_1,\lambda_2,x}$ has a simple closed form for several of the well-known distributions from Table 1; see Table 2. For example, if F_λ is the Poisson(λ) distribution, then $G_{\lambda_1,\lambda_2,x}$ is the Binomial $(x, \lambda_1/(\lambda_1 + \lambda_2))$ distribution.

Table 1: A partial list of convolution-closed distributions. The last two rows contain multivariate distributions. The results in each row are easily verifiable. The generalized Poisson and Tweedie distributions are written in their additive exponential dispersion family parameterization; see Jørgensen and Song (1998) for details.

Distribution	Notes
$X \sim \text{Poisson}(\lambda)$, where $\text{E}[X] = \lambda$ and $\text{Var}(X) = \lambda$.	Convolution-closed in λ.
$X \sim \text{N}(\mu, \sigma^2)$, where $\text{E}[X] = \mu$ and $\text{Var}[X] = \sigma^2$.	Convolution-closed in (μ, σ^2).
$X \sim \text{NegativeBinomial}(r, p)$, where $\text{E}[X] = r\frac{1-p}{p}$ and $\text{Var}[X] = r\frac{1-p}{p^2}$.	Convolution-closed in r if p is fixed.
$X \sim \text{Gamma}(\alpha, \beta)$, where $\text{E}[X] = \frac{\alpha}{\beta}$ and $\text{Var}(X) = \frac{\alpha}{\beta^2}$.	Convolution-closed in α if β is fixed.
$X \sim \text{Binomial}(r, p)$, where $\text{E}[X] = rp$ and $\text{Var}(X) = rp(1-p)$.	Convolution-closed in r if p is fixed.
$X \sim \text{InverseGaussian}(\mu w, \lambda w^2)$ with $\text{E}[X] = \mu w$ and $\text{Var}(X) = \frac{w^3\mu^3}{w^2\lambda} = \frac{w\mu^3}{\lambda}$.	Convolution-closed in w if μ and λ are fixed.
$X \sim \text{GeneralizedPoisson}(\lambda, \theta)$, see Jørgensen and Song (1998) for parameterization.	Convolution-closed in λ if θ is fixed.
$X \sim \text{Tweedie}_p(\lambda, \theta)$, see Jørgensen and Song (1998) for parameterization.	Convolution-closed in λ if θ and p are fixed.
$X \sim \text{N}_k(\mu, \Sigma)$, with $\text{E}[X] = \mu$ and $\text{Var}(X) = \Sigma$.	Convolution-closed in (μ, Σ).
$X \sim \text{Multinomial}_k(r, p)$, with $\text{E}[X] = rp$ and $\text{Var}(X) = r\left(\text{diag}(p) - pp^T\right)$.	Convolution-closed in r if p is fixed.

2.2 Data Thinning

Recall from Section 2.1 that $G_{\lambda_1,\lambda_2,x}$ is the conditional distribution of $X' \mid X' + X'' = x$, where $X' \sim F_{\lambda_1}$ and $X'' \sim F_{\lambda_2}$ with $X' \perp\!\!\!\perp X''$. We now introduce our proposal.

Algorithm 1: Data thinning

Input : A realization x of $X \sim F_\lambda$, where F_λ is convolution-closed in λ with parameter space Λ. A scalar $\epsilon \in (0,1)$ such that $\epsilon\lambda \in \Lambda$ and $(1-\epsilon)\lambda \in \Lambda$.

1 Draw $X^{(1)} \mid X = x \sim G_{\epsilon\lambda,(1-\epsilon)\lambda,x}$.

2 Let $X^{(2)} = X - X^{(1)}$.

Output: $\left(X^{(1)}, X^{(2)}\right)$.

We now introduce our main theorem, which is motivated by a proposal by Joe (1996) to construct autoregressive time series processes with known marginal distributions.

Theorem 5 *Suppose that we apply Algorithm 1 to a realization x of $X \sim F_\lambda$. Then, the following results hold: (i) $X^{(1)} \sim F_{\epsilon\lambda}$ and $X^{(2)} \sim F_{(1-\epsilon)\lambda}$; (ii) $X^{(1)} \perp\!\!\!\perp X^{(2)}$; (iii) If F_λ satisfies the linear expectation property (Definition 2), then $E[X^{(1)}] = \epsilon\, E[X]$ and $E[X^{(2)}] = (1-\epsilon)\, E[X]$.*

Theorem 5 is proven in Appendix A.1. The intuition for parts (i) and (ii) is as follows: if $X \sim F_\lambda$, then X could have arisen as the sum of two independent random variables $X' \sim F_{\lambda_1}$ and $X'' \sim F_{\lambda_2}$, with $\lambda_1 + \lambda_2 = \lambda$. Algorithm 1 works backwards to undo this sum by generating $X^{(1)}$ and $X^{(2)}$ that follow the same distribution as X' and X''. Part (iii) follows from Definition 2. As we will see in Section 4.1, $\epsilon \in (0,1)$ is a tuning parameter that governs a tradeoff between how much information is in $X^{(1)}$ as opposed to $X^{(2)}$.

Theorem 5 guarantees that the decomposition provided by Algorithm 1 satisfies the goals given in Section 1: namely $X = X^{(1)} + X^{(2)}$, $X^{(1)} \perp\!\!\!\perp X^{(2)}$, and $X^{(1)}$ and $X^{(2)}$ follow the same distribution as X, up to a (known) scaling of a parameter. Table 2 summarizes the data thinning proposal for

several well-known distributions. However, Algorithm 1 and Theorem 5 apply well beyond the set of distributions considered in Table 2.

Remark 6 *In Table 2, we focus on distributions where the conditional distribution $G_{\lambda_1,\lambda_2,x}$ has a recognizable form. For distributions where this is not the case, standard numerical sampling algorithms can be used to generate $X^{(1)}$ and $X^{(2)}$, so long as the conditional distribution can be expressed up to a normalizing constant.*

Remark 7 *Some of the decompositions presented in Table 2 require knowledge of an additional parameter that is not of primary interest. For example, thinning the $\mathrm{N}(\mu, \sigma^2)$ distribution requires knowledge of σ^2 (see also: Rasines and Young, 2022; Leiner et al., 2023). In Section 2.3, we explore the implications of performing data thinning in the presence of an unknown nuisance parameter.*

Remark 8 *Table 2 indicates that thinning the $\mathrm{Binomial}(r, p)$ distribution or the $\mathrm{Multinomial}(r, p)$ distribution requires that ϵr take on an integer value. This is because these distributions are not infinitely divisible (Joe, 1996). This restriction becomes more limiting in the extension to multiple folds given in Section 3, and prevents us from thinning the Bernoulli or categorical distributions.*

Remark 9 *The thinning recipe for the Gaussian can be shown to be equivalent, up to a simple rescaling by ϵ, to a procedure for splitting a Gaussian random variable with known variance that has been used in several recent papers (Rasines and Young, 2022; Leiner et al., 2023; Oliveira et al., 2021; Tian and Taylor, 2018; Tian, 2020). We derive this equivalence in Section 5.*

We now give an example of an application where data thinning is useful in practice.

Example 2 (Model validation using data thinning) *Suppose we observe X_{ij} for $i = 1, \ldots, n$ and $j = 1, \ldots, d$, where either each X_{ij} is drawn independently from a univariate convolution-closed distribution that satisfies the linear expectation property from Definition 2, or else each row $(X_{i1}, \ldots, X_{id})^T$ is drawn independently from a multivariate convolution-closed distribution that satisfies the linear expectation property.*

We wish to evaluate $\hat{\mu}(X)$ as an estimator for $E[X]$. Computing a loss function such as mean-squared error between $\hat{\mu}(X)$ and X is unsatisfactory, since the loss function will take on a small value if we overfit the mean. Instead, we apply Algorithm 1 with $\epsilon \in (0, 1)$ to either each element or each row in X, such that each element X_{ij} is thinned into $X^{(1)}_{ij}$ and $X^{(2)}_{ij}$. We compute $\hat{\mu}(X^{(1)})$, which is an estimator of $E[X^{(1)}] = \epsilon\, E[X]$ (Theorem 5, part (iii)). We then compute a loss function between $\hat{\mu}(X^{(1)})$ and $X^{(2)}$. Since $X^{(1)} \perp\!\!\!\perp X^{(2)}$ (Theorem 5, part (ii)), the loss function will not take on small values due to overfitting.

In Example 2, if $\epsilon = 0.5$, then $\mathrm{E}[X^{(1)}] = \mathrm{E}[X^{(2)}] = 0.5\,\mathrm{E}[X]$. Thus, $\hat{\mu}(X^{(1)})$ is an estimator of $\mathrm{E}[X^{(2)}]$, and so devising a suitable loss function is straightforward. If $\epsilon \neq 0.5$, then $\hat{\mu}(X^{(1)})$ is a plug-in estimator of $\frac{\epsilon}{1-\epsilon}\,\mathrm{E}[X^{(2)}]$ (Theorem 5). The following example shows how a loss function that takes into account this factor of ϵ can be constructed in practice.

Example 3 (Example 2 with mean squared error loss) *Suppose we wish to use mean squared error to define a loss function between $\hat{\mu}(X^{(1)})$ and $X^{(2)}$ in Example 2. Since $E[X^{(2)}] = \frac{1-\epsilon}{\epsilon}\, E[X^{(1)}]$, we compute the loss as*

$$\frac{1}{nd}\left\| X^{(2)} - \frac{1-\epsilon}{\epsilon}\hat{\mu}(X^{(1)}) \right\|_F^2,$$

where the factor of $(1-\epsilon)/\epsilon$ turns an estimate of $E[X^{(1)}]$ into an estimate of $E[X^{(2)}]$.

We discuss the choice of ϵ in Section 4.1.

Table 2: Details of data thinning for several well-known distributions, using the parameterizations given in Table 1. While the exponential distribution itself is not convolution-closed in its single parameter, recognizing it as a special case of the gamma distribution with known $\alpha = 1$ yields a decomposition. In all cases, the distribution of $X^{(2)}$ matches that of $X^{(1)}$, with ϵ replaced by $(1-\epsilon)$, with $X^{(1)} \perp\!\!\!\perp X^{(2)}$.

Distribution of X	Generate $X^{(1)} \mid X = x$ as:	Dist. of $X^{(1)}$	Notes
Poisson(λ)	Draw $X^{(1)} \mid X = x \sim \text{Binomial}(x, \epsilon)$.	Poisson($\epsilon\lambda$)	
$\text{N}(\mu, \sigma^2)$	Draw $X^{(1)} \mid X = x \sim \text{N}(\epsilon x, \epsilon(1-\epsilon)\sigma^2)$.	$\text{N}(\epsilon\mu, \epsilon\sigma^2)$	σ^2 must be known.
NegativeBinomial(r, p)	Draw $X^{(1)} \mid X = x \sim \text{BetaBinomial}(x, \epsilon r, (1-\epsilon)r)$.	NegativeBinomial($\epsilon r, p$)	r must be known.
Gamma(α, β)	Draw $Z \sim \text{Beta}(\epsilon\alpha, (1-\epsilon)\alpha)$, and let $X^{(1)} = x \cdot Z$.	Gamma($\epsilon\alpha, \beta$)	α must be known.
Exponential(λ)	Draw $Z \sim \text{Beta}(\epsilon, (1-\epsilon))$, and let $X^{(1)} = x \cdot Z$.	Gamma(ϵ, λ)	
Binomial(r, p)	Draw $X^{(1)} \mid X = x \sim \text{Hypergeometric}(\epsilon r, (1-\epsilon)r, x)$.	Binomial($\epsilon r, p$)	r must be known ϵr must be integer.
$\text{N}_k(\mu, \Sigma)$	Draw $X^{(1)} \mid X = x \sim \text{N}(\epsilon x, \epsilon(1-\epsilon)\Sigma)$.	$\text{N}_k(\epsilon\mu, \epsilon\Sigma)$	Σ must be known.
$\text{Multinomial}_k(r, p)$	Draw $X^{(1)} \mid X = x \sim$ $\text{MultivariateHypergeometric}(x_1, x_2, \ldots, x_k, \epsilon r)$.	$\text{Multinomial}_k(\epsilon r, p)$	r must be known. ϵr must be integer.

2.3 Effect of Unknown Nuisance Parameters

For several of the distributions in Table 2, data thinning requires knowledge of a nuisance parameter. For example, thinning a $\text{N}(\mu, \sigma^2)$ distribution requires knowledge of σ^2.

We now consider what happens when we perform data thinning on Gaussian data using an incorrect value of the variance. We refer to this incorrect value as $\tilde{\sigma}^2$.

Proposition 10 *Suppose that we observe x from $X \sim \text{N}(\mu, \sigma^2)$. We draw $X^{(1)} \mid X = x \sim \text{N}\left(\epsilon x, \epsilon(1-\epsilon)\tilde{\sigma}^2\right)$, for some $\tilde{\sigma}^2$ that is not a function of x, and let $X^{(2)} = X - X^{(1)}$. Then: (i) $X^{(1)} \sim \text{N}\left(\epsilon\mu, \epsilon^2\sigma^2 + \epsilon(1-\epsilon)\tilde{\sigma}^2\right)$, (ii) $X^{(2)} \sim \text{N}\left((1-\epsilon)\mu, (1-\epsilon)^2\sigma^2 + \epsilon(1-\epsilon)\tilde{\sigma}^2\right)$, and (iii) $\text{cov}\left(X^{(1)}, X^{(2)}\right) = \epsilon(1-\epsilon)\left(\sigma^2 - \tilde{\sigma}^2\right)$.*

Part (iii) of Proposition 10 indicates that if we apply data thinning with too little noise ($\tilde{\sigma}^2 < \sigma^2$), then $X^{(1)}$ and $X^{(2)}$ are positively correlated. On the other hand, if we apply data thinning with too much noise ($\tilde{\sigma}^2 > \sigma^2$), then $X^{(1)}$ and $X^{(2)}$ are negatively correlated. Similar results hold for the negative binomial distribution and the gamma distribution.

Proposition 11 *Suppose that we observe x from $X \sim \text{NegativeBinomial}(r, p)$. We draw $X^{(1)} \mid X = x \sim \text{BetaBin}\left(x, \epsilon\tilde{r}, (1-\epsilon)\tilde{r}\right)$ for some $\tilde{r}$ that is not a function of x, and let $X^{(2)} = X - X^{(1)}$. Then $\text{cov}\left(X^{(1)}, X^{(2)}\right) = \epsilon(1-\epsilon)r\left(\frac{1-p}{p}\right)^2\left(1 - \frac{r+1}{\tilde{r}+1}\right)$.*

Proposition 12 *Suppose that we observe x from $X \sim \text{Gamma}(\alpha, \beta)$. We let $X^{(1)} = x \times Z$, where $Z \sim \text{Beta}\left(\epsilon\tilde{\alpha}, (1-\epsilon)\tilde{\alpha}\right)$ for some $\tilde{\alpha}$ that is not a function of x. We let $X^{(2)} = X - X^{(1)}$. Then $\text{cov}\left(X^{(1)}, X^{(2)}\right) = \epsilon(1-\epsilon)\frac{\alpha}{\beta^2}\left(1 - \frac{\alpha+1}{\tilde{\alpha}+1}\right)$.*

Propositions 10–12 are proven in Appendix A.2. Figure 1 verifies these results empirically. The results in this section assume that $\tilde{\sigma}^2$, $\tilde{r}$, and $\tilde{\alpha}$ are not a function of x. In practice, one might estimate the unknown parameters σ^2, r, and α using additional data.

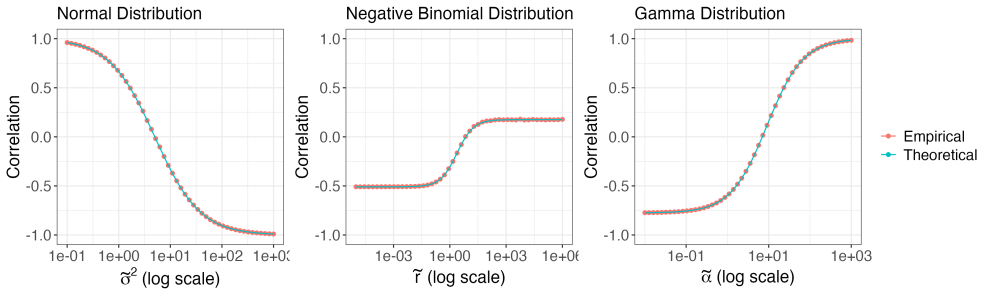

Figure 1: *Left:* We generate 100,000 realizations of $X \sim \mathrm{N}(7, 5)$. For 50 values of $\tilde{\sigma}^2$, we thin X into $X^{(1)}$ and $X^{(2)}$ using $\tilde{\sigma}^2$ instead of $\sigma^2 = 5$. *Center:* We generate 100,000 realizations of $X \sim \mathrm{NB}(7, 0.7)$. For 50 values of $\tilde{r}$, we thin X into $X^{(1)}$ and $X^{(2)}$ using $\tilde{r}$ instead of $r = 7$. *Right:* We generate 100,000 realizations of $X \sim \mathrm{Gamma}(7, 5)$. For 50 values of $\tilde{\alpha}$, we thin X into $X^{(1)}$ and $X^{(2)}$ using $\tilde{\alpha}$ instead of $\alpha = 7$. *All:* In each panel, for each value of the nuisance parameter, we display the empirical correlation between $X^{(1)}$ and $X^{(2)}$ (red dots), along with the theoretical correlation suggested by Propositions 10–12 (blue lines). In all cases, we use $\epsilon = 0.44$ for thinning.

3. Multifold Data Thinning

Data thinning involves decomposing X into $X^{(1)}$ and $X^{(2)}$, which each have the same distribution as X (up to a known parameter scaling). It can be applied recursively to create M independent data folds, $X^{(1)}, \ldots, X^{(M)}$, that sum to X, as in the following example.

Example 4 (Recursive thinning of the normal distribution) *Let x denote a realization of $X \sim \mathrm{N}(\mu, \sigma^2)$. Given $\epsilon_1, \epsilon_2, \epsilon_3 \in (0, 1)$ with $\epsilon_1+\epsilon_2+\epsilon_3 = 1$, we first draw $X^{(1)} \mid X \sim \mathrm{N}\left(\epsilon_1 X, \epsilon_1(1-\epsilon_1)\sigma^2\right)$. Let $X^{(2,3)} = X - X^{(1)}$. By Theorem 5, $\left(X^{(1)}, X^{(2,3)}\right) \sim N\left(\epsilon_1\mu, \epsilon_1\sigma^2\right) \times N\left((1-\epsilon_1)\mu, (1-\epsilon_1)\sigma^2\right)$.*

We next draw $X^{(2)} \mid X^{(2,3)} \sim \mathrm{N}\left(\frac{\epsilon_2}{1-\epsilon_1}X^{(2,3)}, \frac{\epsilon_2}{1-\epsilon_1}(1-\frac{\epsilon_2}{1-\epsilon_1})(1-\epsilon_1)\sigma^2\right)$, and let $X^{(3)} = X - X^{(1)} - X^{(2)}$. By Theorem 5, $\left(X^{(2)}, X^{(3)}\right) \sim \mathrm{N}(\epsilon_2\mu, \epsilon_2\sigma^2) \times N(\epsilon_3\mu, \epsilon_3\sigma^2)$.

Furthermore, since $\left(X^{(2)}, X^{(3)}\right)$ is a function of $X^{(2,3)}$, the pair $\left(X^{(2)}, X^{(3)}\right)$ remains independent of $X^{(1)}$. Thus, $\left(X^{(1)}, X^{(2)}, X^{(3)}\right) \sim N\left(\epsilon_1\mu, \epsilon_1\sigma^2\right) \times N(\epsilon_2\mu, \epsilon_2\sigma^2) \times N(\epsilon_3\mu, \epsilon_3\sigma^2)$.

While Example 4 can be extended to create $M > 3$ folds, this recursive approach can be cumbersome. In Example 1 of Section 1, we saw that, for the gamma distribution, there is a simple way to create multiple folds without recursion. We will now provide a general form of this result. Let $G_{\lambda_1,\lambda_2,\ldots,\lambda_M,x}$ denote the joint distribution of $(X_1, \ldots, X_M) \mid X_1 + X_2 + \ldots + X_M = x$, where $X_m \overset{\text{ind.}}{\sim} F_{\lambda_m}$, for $m = 1, \ldots, M$, and where F_λ is a convolution-closed distribution. The following algorithm and theorem mimic Algorithm 1 and Theorem 5.

Algorithm 2: Multifold data thinning.

Input : A realization x of $X \sim F_\lambda$, where F_λ is a convolution-closed distribution with parameter space Λ. Scalars $\epsilon_1, \ldots, \epsilon_M \in (0, 1)$ such that $\sum_{m=1}^{M} \epsilon_m = 1$ and $\epsilon_m\lambda \in \Lambda$ for $m = 1, \ldots M$.

1 Draw $\left(X^{(1)}, \ldots, X^{(M)}\right) \sim G_{\epsilon_1\lambda,\epsilon_2\lambda,\ldots,\epsilon_M\lambda,x}$.

Output: $\left(X^{(1)}, \ldots, X^{(M)}\right)$

Table 3: Details of how to perform multifold data thinning (Algorithm 2) for several common univariate distributions, where where $\epsilon = (\epsilon_1, \ldots, \epsilon_M)^T$. In the decomposition of the binomial distribution, $\epsilon_m r$ must be an integer. Each row can be verified using properties of these distributions.

Distribution of X	Generate $\left(X^{(1)}, \ldots, X^{(M)}\right) \mid X = X$ as:	Dist. of $X^{(m)}$
Poisson(λ)	$\left(X^{(1)}, \ldots, X^{(M)}\right) \mid X = x \sim \text{Multinomial}(x, \epsilon_1, \ldots, \epsilon_M)$.	Poisson($\epsilon_m \lambda$)
N(μ, σ^2)	$\left(X^{(1)}, \ldots, X^{(M)}\right) \mid X = x \sim \text{N}\left(\mu\boldsymbol{\epsilon}, \sigma^2 \text{diag}(\epsilon) - \sigma^2 \epsilon\epsilon^T\right)$,	N($\epsilon_m \mu, \epsilon_m \sigma^2$),
NegativeBinomial(r, p)	$\left(X^{(1)}, \ldots, X^{(M)}\right) \mid X = x \sim \text{DirichletMultinomial}(X, \epsilon_1 r, \ldots, \epsilon_M r)$.	NegativeBinomial($\epsilon_m r, p$)
Gamma(α, β)	Draw $Z \sim \text{Dirichlet}\left(\epsilon_1 \alpha, \ldots, \epsilon_M \alpha\right)$, and and let $\left(X^{(1)}, \ldots, X^{(M)}\right) = x \cdot Z$	Gamma($\epsilon_m \alpha, \beta$)
Exponential(λ)	Draw $Z \sim \text{Dirichlet}(\epsilon_1, \ldots, \epsilon_M)$, and let $\left(X^{(1)}, \ldots, X^{(M)}\right) = x \cdot Z$.	Gamma(ϵ_m, λ)
Binomial(r, p)	$\left(X^{(1)}, \ldots, X^{(M)}\right) \mid X = x \sim \text{MultivariateHypergeometric}(\epsilon_1 r, \ldots, \epsilon_M r, x)$.	Binomial($\epsilon_m r, p$)

Theorem 13 *Suppose we apply Algorithm 2 to a realization x of $X \sim F_\lambda$, for a convolution-closed distribution F_λ. Then, the following results hold: (i) $X^{(m)} \sim F_{\epsilon_m \lambda}$ for $m = 1, \ldots, M$; (ii) $X^{(1)}, \ldots, X^{(M)}$ are mutually independent; (iii) $X^{(1)} + X^{(2)} + \cdots + X^{(M)} = X$; and (iv) if F_λ satisfies the linear expectation property (Definition 2), then $E[X^{(m)}] = \epsilon_m E[X]$ for $m = 1, \ldots, M$.*

The proof of Theorem 13 is included in Appendix A.1, and is a straightforward extension of that of Theorem 5. The intuition for parts (i)-(iii) is as follows: we know that $X \sim F_\lambda$ could have arisen as the sum of M mutually independent random variables $X_1, \ldots, X_M$ such that $X_m \sim F_{\epsilon_m \lambda}$. If we draw $\left(X^{(1)}, \ldots, X^{(M)}\right) | X = x \sim G_{\epsilon_1\lambda, \epsilon_2\lambda, \ldots, \epsilon_M\lambda, x}$, then the joint distribution of $\left(X^{(1)}, \ldots, X^{(M)}\right)$ equals the joint distribution of $(X_1, \ldots, X_M)$, i.e. it is the joint distribution of M independent random variables with distributions $F_{\epsilon_1\lambda}, \ldots, F_{\epsilon_M\lambda}$. Part (iv) follows directly from Definition 2. We now revisit the case of the Gaussian distribution from Example 4.

Example 5 (Multifold thinning of the normal distribution) *Let $X \sim \text{N}(\mu, \sigma^2)$ and let $\epsilon_1, \epsilon_2, \epsilon_3 > 0$ with $\sum_{i=1}^3 \epsilon_i = 1$. To generate $M = 3$ independent folds of the data, we draw*

$$\begin{bmatrix} X^{(1)} \\ X^{(2)} \\ X^{(3)} \end{bmatrix} \mid X = x \sim N\left(\begin{bmatrix} \epsilon_1 x \\ \epsilon_2 x \\ \epsilon_3 x \end{bmatrix}, \begin{bmatrix} \epsilon_1(1-\epsilon_1)\sigma^2 & -\epsilon_1\epsilon_2\sigma^2 & -\epsilon_1\epsilon_3\sigma^2 \\ -\epsilon_1\epsilon_2\sigma^2 & \epsilon_2(1-\epsilon_2)\sigma^2 & -\epsilon_2\epsilon_3\sigma^2 \\ -\epsilon_1\epsilon_3\sigma^2 & -\epsilon_2\epsilon_3\sigma^2 & \epsilon_3(1-\epsilon_3)\sigma^2 \end{bmatrix}\right).$$

One can verify that this multivariate normal corresponds to $G_{\epsilon_1\lambda, \epsilon_2\lambda, \epsilon_3\lambda, x}$. By Theorem 13, $X^{(1)}, X^{(2)}$, and $X^{(3)}$ are independent and $X^{(m)} \sim \text{N}(\epsilon_m \mu, \epsilon_m \sigma^2)$ for $m = 1, 2, 3$. This distribution $G_{\epsilon_1\lambda, \epsilon_2\lambda, \epsilon_3\lambda, x}$ is a degenerate multivariate normal distribution, which enforces the constraint that the realized values of $X^{(1)}, X^{(2)}$, and $X^{(3)}$ sum to x.

Table 3 reveals that $G_{\epsilon_1\lambda, \epsilon_2\lambda, \ldots, \epsilon_M\lambda, x}$ in Algorithm 2 has a very simple form for every univariate distribution in Table 2. We omit the multivariate distributions to avoid cumbersome notation. Once again, in cases where the conditional distribution is not a recognizable distribution, if its density is known up to a normalizing constant we can generate $X^{(1)}, \ldots, X^{(M)}$ using sampling techniques.

The role of the parameters $\epsilon_1, \ldots, \epsilon_M$ in Algorithm 2 is discussed in Section 4.1. We now consider the following extension of Example 2.

Example 6 (Cross validation using multifold thinning) *In the setting of Example 2, we apply Algorithm 2 with parameters $\epsilon_1, \ldots, \epsilon_M$ to either each element or each row of X such that each element X_{ij} is thinned into $X_{ij}^{(1)}, \ldots, X_{ij}^{(M)}$.*

Then, for $m = 1, \ldots, M$, we first define $X^{(-m)} := X - X^{(m)}$. We obtain $\hat{\mu}\left(X^{(-m)}\right)$, which is an estimator of $E[X^{(-m)}] = (1 - \epsilon_m) E[X]$. We then compute a loss function between $\hat{\mu}\left(X^{(-m)}\right)$ and $X^{(m)}$. For example, as in Example 3, we can compute the mean squared error between $\frac{\epsilon_m}{1-\epsilon_m}\hat{\mu}\left(X^{(-m)}\right)$ and $X^{(m)}$. We evaluate the estimator $\hat{\mu}(\cdot)$ by averaging the loss across folds.

The advantage of multifold thinning (Example 6) over single fold thinning with $\epsilon = 1/M$ (Example 2) is reduction of the variance of the loss function via averaging. We will demonstrate the practical advantages of multifold thinning in Section 6.

4. Comparing Data Thinning and Sample Splitting

In comparing data thinning and sample splitting in a particular setting, there are two considerations. First, we must figure out if each method is applicable. Then, in settings where both are applicable, we must figure out if one method is preferable.

Data thinning requires an assumption that each entry in our data set is drawn from a specific (but possibly different) convolution-closed distribution. Sample splitting requires no such parametric assumption. On the other hand, we cannot apply sample splitting if our task requires estimating a parameter for every individual observation or drawing conclusions about specific observations. For example, suppose that we wish to evaluate the performance of a clustering algorithm. After sample splitting, clustering the observations in the training set does not yield cluster assignments for the observations in the test set, and thus there is nothing to evaluate on the test set (see e.g. Gao et al., 2022; Fu and Perry, 2020). Similarly, it is not clear how to use sample splitting to validate a low-rank matrix approximation, for which a latent coordinate must be estimated for each of the n observations (see, e.g. Owen and Perry, 2009). We focus on these examples, where sample splitting is not an option, in Sections 6 and 7.

In Section 4.1, we show that the parameters $\epsilon_1, \ldots, \epsilon_M$ in multifold data thinning (Algorithm 2) control the allocation of information across folds. In Section 4.2, we use this information allocation result to theoretically argue that sample splitting and data thinning achieve similar performance, but that data thinning may be preferable in settings where the observations are not identically distributed. In particular, we see that sample splitting is not an attractive choice for "fixed-covariate" regression in the presence of high-leverage points. In Section 4.3, we empirically compare data thinning and sample splitting in a setting where both are applicable.

4.1 Role of the Parameters $\epsilon_1, \ldots, \epsilon_M$ in Multifold Data Thinning

In Algorithm 2, the parameters $\epsilon_1, \ldots, \epsilon_M$ determine how the information in a random variable X about an unknown parameter is allocated across folds of data.

Theorem 14 *Suppose that we thin a random variable X using Algorithm 2 with parameters $\epsilon_1, \ldots, \epsilon_M$ to obtain $X^{(1)}, \ldots, X^{(M)}$. Let $I_X(\theta)$ denote the Fisher information contained in X about an unknown parameter θ, i.e. a parameter in the distribution of X that does not appear in $G_{\epsilon_1\lambda,\ldots,\epsilon_M\lambda,x}$. Assume that this Fisher information exists. Then $I_{X^{(m)}}(\theta) = \epsilon_m I_X(\theta)$ for $m = 1, \ldots, M$.*

Remark 15 *In the context of Theorem 14, the parameter θ may or may not be a function of the convolution-closed parameter λ, but it must be a parameter that is unknown during the thinning process. Below, we list a few examples where Theorem 14 applies and where it does not.*

- ***Poisson distribution:*** *Let $X \sim \text{Poisson}(\lambda)$, and suppose we thin X to obtain $X^{(m)} \sim \text{Poisson}(\epsilon_m\lambda)$. As λ is unknown during the thinning process, Theorem 14 says that $I_{X^{(m)}}(\lambda) = \epsilon_m I_X(\lambda)$. We can easily verify this by direct calculation: $I_X(\lambda) = \frac{1}{\lambda}$ and $I_{X^{(m)}}(\lambda) = \frac{\epsilon_m}{\lambda}$.*

- ***Binomial distribution:*** *Let* $X \sim \text{Binomial}(r, p)$*, and suppose that we thin* X *to obtain* $X^{(m)} \sim \text{Binomial}(\epsilon_m r, p)$*. Then* $I_X(p) = r/(p(1-p))$*, and direct calculation verifies that* $I_{X^{(m)}}(p) = \epsilon_m I_X(p)$*. On the other hand, Theorem 14 makes no claims about the parameter* r*, since* r *must be known during thinning.*

- ***Gaussian distribution:*** *Let* $X \sim \text{N}(\mu, \sigma^2)$*, and suppose that we thin* X *to obtain* $X^{(m)} \sim \text{N}(\epsilon_m \mu, \epsilon_m \sigma^2)$*. Then direct computation verifies that* $I_{X^{(m)}}(\mu) = \epsilon_m I_X(\mu)$*. However, Theorem 14 makes no claims about the parameter* σ^2*, since it must be known during thinning.*

Theorem 14 implies that when choosing $\epsilon_1, \ldots, \epsilon_M$ for Algorithm 2 or when choosing ϵ for Algorithm 1, one should consider how much information to devote to the training task (i.e. model fitting) as opposed to the testing task (i.e. model evaluation). In Example 2, as ϵ increases, the quality of the estimator $\hat{\mu}(X^{(1)})$ increases, but the information available for computing the loss between this estimator and $X^{(2)}$ decreases.

Recall that applying Algorithm 1 to X with parameter $\epsilon = \frac{1}{M}$ yields $X^{(1)}$ and $X^{(2)}$ with the same distributions as $X^{(m)}$ and $X - X^{(m)}$ when we apply Algorithm 2 to X with $\epsilon_1 = \ldots = \epsilon_M = \frac{1}{M}$. Thus, the information allocation between $X^{(m)}$ and $X - X^{(m)}$ after multifold thinning is the same as the information allocation between $X^{(1)}$ and $X^{(2)}$ after two-fold thinning with $\epsilon = \frac{1}{M}$. The advantage of using multiple folds for model validation (i.e. Example 6 rather than Example 2) comes from the reduction in variance that results from averaging the loss function across folds.

4.2 Theoretical Comparison to Sample Splitting

In this section, we focus on the case of two-fold thinning (Algorithm 1) for simplicity. Understanding the way in which the parameter ϵ in Algorithm 1 partitions information about parameters of interest helps us draw direct connections between data thinning (with parameter ϵ) and sample splitting (with ϵ denoting the proportion of observations assigned to the training set).

Corollary 16 *Suppose that we observe* n *independent and identically distributed (iid) random variables* $\mathbf{X} = (X_1, \ldots, X_n)$*, where* $X_i \sim F_\lambda$*. Assume that* ϵn *is an integer and that* $\epsilon\lambda \in \Lambda$*. Consider the following two methods for splitting* $\mathbf{X}$ *into independent training and test sets.*

- ***Sample splitting:*** *Assign a specific set of* ϵn *observations to the training set, denoted* $\mathbf{X}_{\text{ss}}^{\text{train}}$*, and the remaining* $(1-\epsilon)n$ *observations to the test set, denoted* $\mathbf{X}_{\text{ss}}^{\text{test}}$*.*

- ***Data thinning:*** *For* $i = 1, \ldots, n$*, thin* X_i *into* $X_i^{(1)}$ *and* $X_i^{(2)}$ *by applying Algorithm 1 with parameter* ϵ*. Let* $\mathbf{X}_{\text{dt}}^{\text{train}} = \left(X_1^{(1)}, \ldots, X_n^{(1)}\right)$ *be the training set and let* $\mathbf{X}_{\text{dt}}^{\text{test}} = \left(X_1^{(2)}, \ldots, X_n^{(2)}\right)$ *be the test set.*

Let θ *be an unknown parameter of interest, as in Theorem 14. Then,* $I_{\mathbf{X}_{\text{ss}}^{\text{train}}}(\theta) = I_{\mathbf{X}_{\text{dt}}^{\text{train}}}(\theta) = \epsilon I_{\mathbf{X}}(\theta)$*, and* $I_{\mathbf{X}_{\text{ss}}^{\text{test}}}(\theta) = I_{\mathbf{X}_{\text{dt}}^{\text{test}}}(\theta) = (1-\epsilon) I_{\mathbf{X}}(\theta)$*.*

In Corollary 16, $I_{\mathbf{X}_{\text{ss}}^{\text{train}}}(\theta)$ takes on the same value for any specific allocation of datapoints to the training set. This means that if we instead randomly allocate data points to the training set, as is typical in practice, the information split remains identical to data thinning. Consequently, we expect sample splitting and data thinning to have similar performance regarding inference on unknown parameters in settings where both are options and where the datapoints are independent and identically distributed. A similar point was made by Leiner et al. (2023), who view their data fission technique as a "continuous analog" of sample splitting, since the requirement for sample splitting that ϵn be an integer limits the choice of ϵ, especially when n is small.

We next consider a setting where the observations are independent but not identically distributed, and thus the difference between data thinning and sample splitting is more pronounced.

Example 7 (Fixed-covariate regression) *Suppose that we observe a fixed set of covariates $Z_1, \ldots, Z_n$. Suppose that $X_i \overset{\text{ind.}}{\sim} N\left(\beta Z_i, \sigma^2\right)$ for $i = 1, \ldots, n$, and let $\mathbf{X} = (X_1, \ldots, X_n)$. In this setting, $I_{X_i}(\beta) = \frac{Z_i^2}{\sigma^2}$, meaning that observations with larger values of Z_i^2 (high-leverage points) contain more information about β, the unknown parameter of interest. Let $I_{\mathbf{X}_{\text{ss}}^{\text{train}}}(\beta)$ and $I_{\mathbf{X}_{\text{dt}}^{\text{train}}}(\beta)$ be defined as in Corollary 16, where the unknown parameter of interest is the slope β. Then*

$$I_{\mathbf{X}_{\text{dt}}^{\text{train}}}(\beta) = \sum_{i=1}^{n} I_{X_i^{(1)}}(\beta) = \sum_{i=1}^{n} \epsilon \frac{Z_i^2}{\sigma^2} = \epsilon I_{\mathbf{X}}(\beta).$$

However,

$$I_{\mathbf{X}_{\text{ss}}^{\text{train}}}(\beta) = \sum_{i \in \text{train}} I_{X_i}(\beta) = \sum_{i \in \text{train}} \frac{Z_i^2}{\sigma^2} \neq \epsilon I_{\mathbf{X}}(\beta),$$

where train *denotes the specific indices of the observations assigned to the training set. Thus, while data thinning always allocates a fraction ϵ of the Fisher information to the training set, the information allocation of sample splitting depends on the specific assignment of observations to the training set.*

Example 7 shows that $I_{\mathbf{X}_{\text{ss}}^{\text{train}}(\beta)} \neq I_{\mathbf{X}_{\text{dt}}^{\text{train}}(\beta)}$ for a particular assignment of observations to the training set. However, when we perform sample splitting, we typically randomly allocate datapoints to the training set. Under such a procedure, $I_{\mathbf{X}_{\text{ss}}^{\text{train}}(\theta)}$ from Example 7 becomes a random variable that depends on the particular split of the data. If all permissible random splits of the data are equally likely, then $E\left[I_{\mathbf{X}_{\text{ss}}^{\text{train}}(\theta)}\right] = I_{\mathbf{X}_{\text{dt}}^{\text{train}}(\theta)}$, where the expected value is taken over all permissible splits of the data. The same equality holds for the test set, i.e. $E\left[I_{\mathbf{X}_{\text{ss}}^{\text{test}}(\theta)}\right] = I_{\mathbf{X}_{\text{dt}}^{\text{test}}(\theta)}$. Despite this equivalence in the *expected* information allocation, Proposition 1 from Rasines and Young (2022) provides clever insight that tells us why we might prefer the non-random information allocation of data thinning in this setting. Remark 17 instantiates the general result of Rasines and Young (2022) to the simple setting of Example 7.

Remark 17 (Effect on confidence interval width) *In the context of Example 7, suppose that our ultimate goal involves forming a confidence interval for β using the test set. If we are using a maximum-likelihood estimator for β, then the width of the confidence interval computed on the test set under sample splitting and data thinning are proportional, respectively, to the square roots of $\frac{1}{I_{\mathbf{X}_{\text{ss}}^{\text{test}}(\beta)}}$ and $\frac{1}{I_{\mathbf{X}_{\text{dt}}^{\text{test}}(\beta)}}$. Jensen's inequality yields the following result:*

$$E\left[\frac{1}{I_{\mathbf{X}_{\text{ss}}^{\text{test}}}(\beta)}\right] \geq \frac{1}{E\left[I_{\mathbf{X}_{\text{ss}}^{\text{test}}}(\beta)\right]} = \frac{1}{I_{\mathbf{X}_{\text{dt}}^{\text{test}}(\beta)}}.$$

Thus, the confidence intervals for β computed using sample splitting will be wider, on average, than those computed using data thinning in the setting where our observations are not identically distributed. A natural corollary is that sample splitting will achieve lower power than data thinning in this setting, as we will see in Section 4.3.

4.3 Empirical Comparison to Sample Splitting

We now show empirically that sample splitting and data thinning achieve comparable performance in settings where both are applicable and where the observations are independent and identically distributed.

We let $n = 100$ and $p = 20$, and we generate $10,000$ realizations of $\mathbf{Z} \in \mathbb{R}^{n \times p}$, where the entries of $\mathbf{Z}$ are drawn independently from $N(0,1)$. For each realization, we then generate $\mathbf{X} \mid \mathbf{Z} \sim N_n(\mathbf{Z}\beta, I_n)$, where $\beta_1 = \beta_2 = \ldots = \beta_5 = \beta^*$ and $\beta_6 = \beta_7 = \ldots = \beta_{20} = 0$. Our goal is to perform model selection to identify the covariates with non-zero entries in β, and then to form confidence intervals for the coefficients of the selected covariates.

As we cannot naively use the entire data set $(\boldsymbol{Z}, \boldsymbol{X})$ to do both model selection and inference, we consider the following approach.

Step 1: Split the data into a training set and a test set.

Step 2: Perform forward stepwise regression on the training set to select a model that includes some subset of the $p = 20$ covariates. We use the `R` function `step` with its default settings.

Step 3: Re-fit the selected model using the test set and report the standard confidence intervals for each selected coefficient.

We carry out this process using two different methods.

Sample splitting: Randomly generate a set train $\subset \{1, \ldots, n\}$ with $|\text{train}| = \epsilon n$. Let $\{(\boldsymbol{Z}_i, X_i) : i \in \text{train}\}$ be the training set and let $\{(\boldsymbol{Z}_i, X_i) : i \in \{1, \ldots, n\} \setminus \text{train}\}$ be the test set.

Data thinning: Apply Algorithm 1 to each X_i for $i = 1, \ldots, n$. Let $\left\{\left(\mathbf{Z}_i, X_i^{(1)}\right) : i \in \{1, \ldots, n\}\right\}$ be the training set and let $\left\{\left(\mathbf{Z}_i, X_i^{(2)}\right) : i \in \{1, \ldots, n\}\right\}$ be the test set.

We carry out each method using $\epsilon = 0.2, \epsilon = 0.5$, and $\epsilon = 0.8$. For each method, we consider (i) *detection*: the proportion of data sets for which Z_3 appears in the selected model, and (ii) *power*: the proportion of data sets for which the confidence interval for β_3 does not include 0, among those where Z_3 appeared in the selected model. These metrics focus on one of the important covariates, Z_3, but the results are similar for each of the important covariates (i.e. Z_j for $j \in \{1, \ldots, 5\}$). The results are shown in the top row of Figure 2. As expected, data thinning and sample splitting achieve nearly identical results for these two metrics, with ϵ governing a tradeoff between detection and power.

We then repeat the experiment, but we let $n = 42$ and we let $\mathbf{Z}$ be a fixed matrix that contains a single row whose elements are drawn from $N(5,1)$ rather than $N(0,1)$. This corresponds to having a single high-leverage observation in the data set that contains most of the information about β. We only consider $\epsilon \geq 0.5$, since in this setting where $n \approx 2p$, using a smaller value of ϵ would complicate our ability to fit a linear regression model on the training set. As shown in the bottom row of Figure 2, data thinning outperforms sample splitting in this setting, because for any particular split, sample splitting either leaves very little information in the training set or very little in the test set. The power results confirm the insight from Remark 17.

The general finding that data thinning outperforms sample splitting in this setting mirrors findings from Leiner et al. (2023) and Rasines and Young (2022), which is not surprising in light of Remark 9, which states that Gaussian data thinning proposal is equivalent (up to a simple rescaling) to the Gaussian randomization or fission proposals of these prior papers. Thus, the similar conclusions are not a coincidence, and the properties of data thinning seen in this setting can also be interpreted as properties of (Gaussian) data fission.

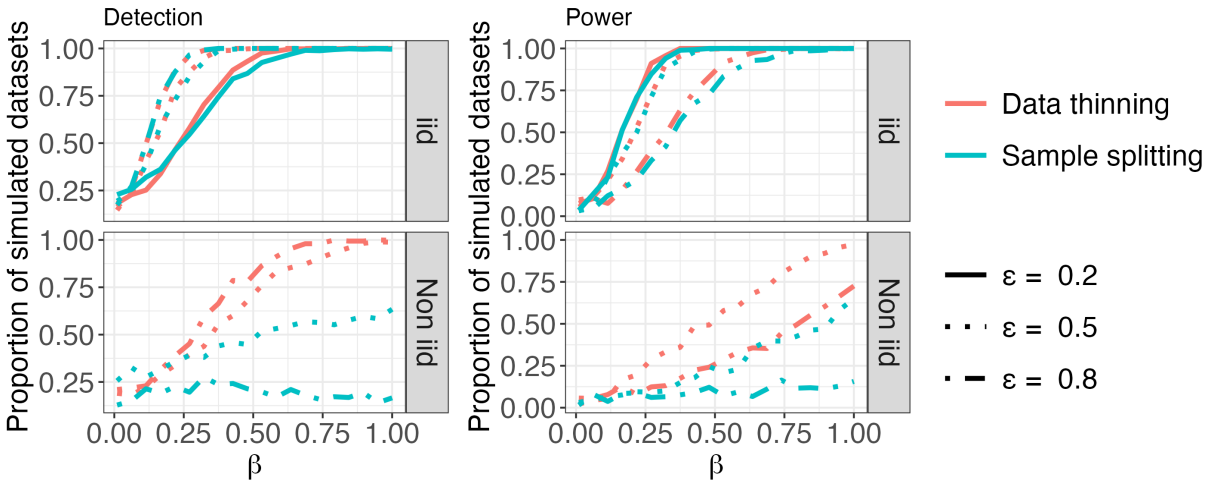

Figure 2: Comparison of data thinning and sample splitting, using the detection and power metrics defined in Section 4.3. The top row shows the results of the large n setting where the observations are independent and identically distributed (iid), and thus data thinning and sample splitting achieve nearly identical results across values of ϵ. The bottom row shows the results in the small n setting, in which the observations are not identically distributed (non iid) and the presence of a high leverage point causes data thinning to outperform sample splitting.

5. Comparing Data Thinning and Data Fission

As mentioned in Section 1, the data fission proposal of Leiner et al. (2023) provides an alternate set of strategies to decompose a single realization X into $X^{(1)}$ and $X^{(2)}$. In this section, we compare and contrast the two approaches.

5.1 Independent Decompositions

Leiner et al. (2023) provide a strategy for obtaining independent $X^{(1)}$ and $X^{(2)}$ only in the case where X is Poisson or Gaussian.

In the case of the Poisson distribution, the proposal of Leiner et al. (2023) coincides exactly with the proposal obtained from Algorithm 1 in this paper. This proposal follows from a classical property of the Poisson distribution (see e.g. (Durrett, 2019), Section 3.7.2), and has recently been applied in contexts related to that of this paper by Oliveira et al. (2022); Sarkar and Stephens (2021); Gerard (2020); Chen et al. (2021) and Neufeld et al. (2022).

In the case of the Gaussian distribution, the proposal of Leiner et al. (2023) has also been used by Tian and Taylor (2018), Oliveira et al. (2021), and Rasines and Young (2022), among others. It does not follow directly from Algorithm 1 in this paper, since $X \neq X^{(1)} + X^{(2)}$. However, in Example 8, we show the proposal of Leiner et al. (2023) is a simple rescaling of the proposal in this paper.

Example 8 (Comparison of two Gaussian decompositions) *Consider the task of splitting the* $\mathrm{N}(\mu, \sigma^2)$ *distribution into two independent normally-distributed random variables, with* σ *known.*

The data thinning proposal is given in Table 2, and leads to $X^{(1)} \sim \mathrm{N}(\epsilon\mu, \epsilon\sigma^2)$ *and* $X^{(2)} \sim \mathrm{N}((1-\epsilon)\mu, (1-\epsilon)\sigma^2)$*, where* $X^{(1)} + X^{(2)} = X$*.*

The data fission proposal is as follows: given a value of $\tau > 0$*, we draw* $Z \sim \mathrm{N}(0, \sigma^2)$*, and then let* $X^{(1)} = X + \tau Z$ *and* $X^{(2)} = X - \frac{1}{\tau}Z$*. Then,* $X' \sim \mathrm{N}(\mu, (1+\tau^2)\sigma^2)$ *and* $X'' \sim \mathrm{N}(\mu, (1+\frac{1}{\tau^2})\sigma^2)$*, with* $X' \perp\!\!\!\perp X''$*. Under this decomposition,* $X' + X'' \neq X$*, but*

$$\frac{1}{1+\tau^2}X' + \frac{\tau^2}{1+\tau^2}X'' = \frac{1}{1+\tau^2}(X + \tau Z) + \frac{\tau^2}{1+\tau^2}(X - \frac{1}{\tau}Z) = X.$$

We can easily verify that, if we let $\epsilon = \frac{1}{1+\tau^2}$*, then the random variables* $\frac{1}{1+\tau^2}X'$ *and* $\frac{\tau^2}{1+\tau^2}X''$ *obtained via data fission have the same distributions (both marginally and conditional on* $X = x$*) as* $X^{(1)}$ *and* $X^{(2)}$ *obtained via data thinning. For example, we can easily verify that* $\frac{1}{1+\tau^2}X' \sim \mathrm{N}\left(\frac{1}{1+\tau^2}\mu, \frac{1}{1+\tau^2}\sigma^2\right) = \mathrm{N}\left(\epsilon\mu, \epsilon\sigma^2\right)$*. Thus, the two decompositions are identical, up to a scaling of* $X^{(1)}$ *and* $X^{(2)}$ *by a (known) constant.*

The main idea of Example 8 extends to the decomposition of the multivariate normal given in Table 2 of this paper, and the corresponding decomposition from Leiner et al. (2023).

5.2 Non-independent Decompositions

With the exception of the Gaussian and Poisson distributions, the decompositions of Leiner et al. (2023) do not yield $X^{(1)}$ and $X^{(2)}$ that are independent. Instead, the goal of Leiner et al. (2023) is to obtain a decomposition such that the distributions of $X^{(1)}$ and $X^{(2)} \mid X^{(1)}$ are tractable. While in principle we can fit a model to $X^{(1)}$ and validate it using the conditional distribution of $X^{(2)} \mid X^{(1)}$, we will see in this section that this can be difficult to carry out in practice. In particular, we note the following drawbacks of the non-independent decompositions of Leiner et al. (2023).

(1) The distribution of $X^{(1)}$, and the conditional distribution of $X^{(2)}|X^{(1)}$, need not resemble the distribution of X. Thus, if the goal is to evaluate a potential model for X, it is not always clear what model to fit to $X^{(1)}$. We will illustrate this drawback in Example 9.

(2) The parameters of interest are entangled in the conditional distribution of $X^{(2)} \mid X^{(1)}$. We will illustrate this issue in Example 10.

(3) The tuning parameter that governs the information trade-off between $X^{(1)}$ and $X^{(2)}$ can be hard to interpret. For instance, in the case of the gamma decomposition in Example 9, the tuning parameter is $B \in \{1, 2, \ldots\}$, but in the case of the negative binomial distribution in Example 10, it is $\epsilon \in (0, 1)$. These both contrast with Example 8, where the tuning parameter was $\tau > 0$. Whereas our Theorem 14 guarantees a specific, pre-specified allocation of Fisher information between $X^{(1)}$ and $X^{(2)}$ after data thinning, the allocation of Fisher information between $X^{(1)}$ and $X^{(2)} \mid X^{(1)}$ after data fission can depend on the value of unknown parameters. We will illustrate this final point in Example 9.

(4) The roles of $X^{(1)}$ and $X^{(2)}$ cannot be interchanged. For example, in the decomposition of the Bernoulli(θ) distribution given in Leiner et al. (2023), the distributions of $X^{(1)}$ and $X^{(2)} \mid X^{(1)}$ each contain information about θ. However, the distribution of $X^{(1)} \mid X^{(2)}$ contains no information about θ. Furthermore, while Remark 1 in Leiner et al. (2023) provides a strategy for obtaining multiple folds of training data for the data fission decompositions that are constructed using the "conjugate prior" strategy, these folds are not marginally

independent of one another (they are conditionally independent given X). Beyond these specific decompositions, Leiner et al. (2023) do not provide a clear strategy for extending their decompositions to the case of multiple folds. Thus, it is not clear in general how to use data fission decompositions to carry out cross validation.

To illustrate points (1) and (3), we consider the gamma distribution.

Example 9 (Gamma decomposition, data fission approach) *Suppose* $X \sim \text{Gamma}(\alpha, \beta)$. *For a tuning parameter* $B \in \{1, 2, \ldots\}$, *Leiner et al. (2023) propose drawing* $Z = (Z_1, \ldots, Z_B)$, *where* $Z_i \overset{\text{ind.}}{\sim} \text{Poisson}(X)$, *and thus each* Z_i *marginally follows a* $\text{NegativeBinomial}(\alpha, \beta/(\beta+1))$ *distribution, and the* Z_i *are independent conditional on* X. *Take* $X^{(1)} = Z$, *and* $X^{(2)} = X$. *Then, the conditional distribution of* $X^{(2)} \mid X^{(1)}$ *is* $\text{Gamma}(\alpha + \sum_{i=1}^{B} Z_i, \beta + B)$. *Furthermore, while* $I_X(\beta) = \frac{\alpha}{\beta^2}$, *direct calculation yields that* $I_{X^{(2)}|X^{(1)}}(\beta) = \frac{\alpha}{\beta^2 + \beta B}$.

This stands in notable contrast to Example 1 in Section 1, in which data thinning provides independent (and gamma-distributed) random variables. Given that $X^{(1)}$ does not resemble X, it is not clear how to apply this decomposition in the setting of Example 12 from Section 6, where we would like to apply a clustering algorithm to $X^{(1)}$ to estimate the true cluster structure of X. Unless $B = 1$, $X^{(1)}$ and X do not even have the same dimensions, which makes it difficult to know what type of clustering algorithm to apply or how to interpret the results. Furthermore, the proportion of Fisher information about β in X that is allocated to the test set $X^{(2)} \mid X^{(1)}$ in Example 9 depends not only on the tuning parameter B, but also on the value of the unknown parameter β. This is in contrast to data thinning, where Theorem 14 guarantees that this proportion will always be equal to the tuning parameter ϵ_2.

To illustrate point (2), we revisit Example 2 to see a concrete example in which data thinning is straightforward but the proposal of Leiner et al. (2023) is difficult to use in practice.

Example 10 (A comparison of negative binomial decompositions) *We observe* $X_{ij} \sim \text{NegativeBinomial}\,(r_{ij}, p_{ij})$ *for* $i = 1, \ldots, n$ *and* $j = 1, \ldots, p$. *Our goal is to evaluate a function* $\hat{\mu}(X)$ *as an estimator for* $E[X]$, *where* $E[X]_{ij} = r_{ij}\frac{1-p_{ij}}{p_{ij}}$.

From Table 2, data thinning requires r_{ij} *to be known, and yields* $X^{(1)}_{ij} \sim \text{NegativeBinomial}\,(\epsilon r_{ij}, p_{ij})$ *and* $X^{(2)}_{ij} \sim \text{NegativeBinomial}\,((1-\epsilon) r_{ij}, p_{ij})$, *with* $X^{(1)} \perp\!\!\!\perp X^{(2)}$, $E[X^{(1)}] = \epsilon\, E[X]$, *and* $E[X^{(2)}] = (1-\epsilon)\, E[X]$. *Thus, as in Example 2 and Example 3,* $\hat{\mu}\left(X^{(1)}\right)$ *is an estimator for* $\epsilon\, E[X]$, *and we can evaluate* $\hat{\mu}(X^{(1)})$ *by computing the mean squared error between* $X^{(2)}$ *and* $\frac{1-\epsilon}{\epsilon}\hat{\mu}\left(X^{(1)}\right)$.

For $\epsilon \in (0, 1)$, *the data fission proposal of Leiner et al. (2023) draws* $X^{(1)}_{ij} \mid X_{ij} \sim \text{Binomial}(X_{ij}, \epsilon)$ *and sets* $X^{(2)} = X - X^{(1)}$. *Under this decomposition,* $X^{(1)}_{ij} \sim \text{NegativeBinomial}\left(r_{ij}, \frac{p_{ij}}{p_{ij}+\epsilon(1-p_{ij})}\right)$, *and so* $E[X^{(1)}] = \epsilon\, E[X]$. *Although marginally* $E[X^{(2)}] = (1-\epsilon)\, E[X]$, *as* $X^{(1)}$ *and* $X^{(2)}$ *are not independent, we cannot simply use mean squared error loss between* $X^{(2)}$ *and* $\frac{1-\epsilon}{\epsilon}\hat{\mu}\left(X^{(1)}\right)$ *to evaluate the estimator. Instead, we must construct a loss function that evaluates* $\hat{\mu}\left(X^{(1)}\right)$ *as an estimator of* $E[X]$ *in the conditional distribution of* $X^{(2)} \mid X^{(1)}$, *which is given by* $X^{(2)}_{ij} \mid X^{(1)}_{ij} \sim \text{NegativeBinomial}\left(r_{ij} + X^{(1)}_{ij}, p_{ij} + \epsilon - p_{ij}\epsilon\right)$. *As* $E[X^{(2)}_{ij} \mid X^{(1)}_{ij}] = \left(r_{ij} + X^{(1)}_{ij}\right)\left(\frac{1-p_{ij}-\epsilon+\epsilon p_{ij}}{p_{ij}+\epsilon-\epsilon p_{ij}}\right)$ *is not a simple function of* $E[X]$, *this is not a straightforward task. At a minimum, it involves disentangling the roles of the parameters* r_{ij} *and* p_{ij}.

Furthermore, while at first glance it might appear that an advantage of data fission over data thinning is that the former does not require knowledge of r_{ij} *to obtain* $X^{(1)}_{ij}$ *and* $X^{(2)}_{ij}$, *evaluating an*

estimator of $E[X]$ using this conditional distribution will require knowing or accurately estimating the nuisance parameters r_{ij} due to the aforementioned parameter entanglement issue.

Similar issues arise for other decompositions given in Leiner et al. (2023). For example, the data fission decomposition of the binomial distribution yields a complicated unnamed distribution for $X^{(2)} \mid X^{(1)}$, which would be very difficult to use in the context of Example 11.

6. Simulation Study

6.1 Simulation Setup

In this section, we focus on the application of data thinning to cross-validation in two settings. We contrast its performance to naive approaches that use the same data to both fit and validate the models. Specifically, we consider Examples 11 and 12. In each of these examples, sample splitting is not a viable option, as the parameters of interest have dimensions equal to the number of observations. Furthermore, as pointed out in Section 5.2, applying data fission in these settings is not straightforward.

Example 11 (Choosing the number of principal components on binomial data) *We generate data with $n = 250$ observations and $d = 100$ dimensions. Specifically, for $i = 1, \ldots, n$ and $j = 1, \ldots, d$, we generate $X_{ij} \overset{\text{ind.}}{\sim} \text{Binomial}(r, p_{ij})$ where $r = 100$ and p is an unknown $n \times d$ matrix of probabilities. We construct $\text{logit}(p)$ as a rank-$K^* = 10$ matrix with singular values $5, 6, \ldots, 14$. Additional details are provided in Section D. Our goal is to estimate K^*.*

Example 12 (Choosing the number of clusters on gamma data) *We generate data sets $X \in \mathbb{R}^{n \times d}$ such that there are 100 observations in each of K^* clusters, for a total of $n = 100K^*$ observations. Our objective is to estimate K^*. We let $X_{ij} \overset{\text{ind.}}{\sim} \text{Gamma}(\lambda, \theta_{c_i,j})$, for $i = 1, \ldots, n$ and $j = 1, \ldots, d$, where $c_i \in \{1, 2, \ldots, K^*\}$ indexes the true cluster membership of the ith observation. The shape parameter λ is a known constant common across all clusters and all dimensions, whereas the rate parameter θ is an unknown $K^* \times d$ matrix such that each cluster has its own d-dimensional rate parameter. We generate data under two regimes: (1) a small d, small K^* regime in which $d = 2$ and $K^* = 4$, and (2) a large d, large K^* regime in which $d = 100$ and $K^* = 10$. The values of λ and θ are provided in Section D. A sample "small d, small K^*" data set is presented in Figure 3, alongside the output of data thinning with $\epsilon = 0.5$.*

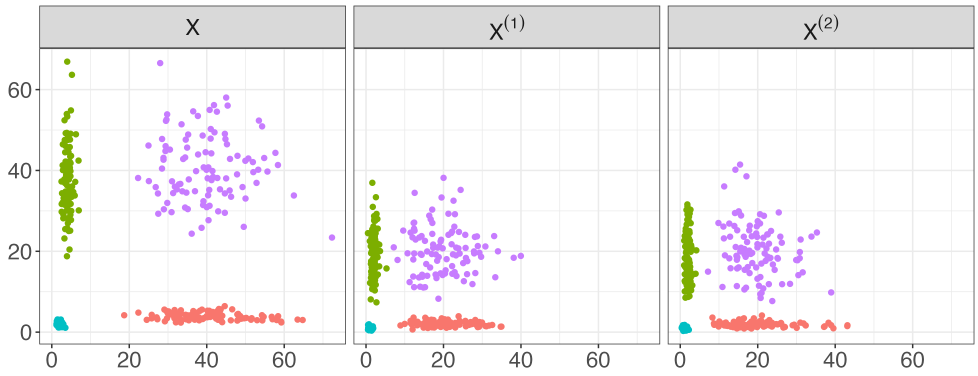

Figure 3: *Left*: A simulated data set in the $d = 2$, $K^* = 4$ setting described in Example 12. *Center/Right*: The result of data thinning with $\epsilon = 0.5$.

6.2 Methods

We use Algorithm 3 to select the number of principal components in binomial data, as in Example 11, using data thinning.

Algorithm 3: Evaluating binomial principal components with negative log-likelihood loss

Input : A positive integer K, a matrix $X \in \mathbb{Z}_{[0,r]}^{n \times d}$, where $X_{ij} \overset{\text{ind.}}{\sim} \text{Binomial}(r, p_{ij})$, and positive scalars $\epsilon^{(\text{train})}$ and $\epsilon^{(\text{test})} = 1 - \epsilon^{(\text{train})}$ such that $\epsilon^{(\text{train})} r, \epsilon^{(\text{test})} r \in \mathbb{Z}_{>0}$.

1 Apply data thinning to X to obtain $X^{(\text{train})}$ and $X^{(\text{test})}$, where $X_{ij}^{(\text{train})} \overset{\text{ind.}}{\sim} \text{Binomial}\left(\epsilon^{(\text{train})} r, p_{ij}\right)$ and $X_{ij}^{(\text{test})} \overset{\text{ind.}}{\sim} \text{Binomial}\left(\epsilon^{(\text{test})} r, p_{ij}\right)$.

2 Compute the singular value decomposition of the log-odds of $X^{(\text{train})}$,

$$\hat{U}\hat{D}\hat{V}^T = \text{logit}\left\{(X^{(\text{train})} + 0.001)/(\epsilon^{(\text{train})} r + 0.002)\right\}.$$

Pseudo-counts prevent taking the logit of 0 or 1.

3 Construct the rank-K approximation of $X^{(\text{train})}$, $p^{(K)} := \text{expit}\left(\hat{U}_{1:K}\hat{D}_{1:K}\hat{V}_{1:K}\right)$.

4 Compute the negative log-likelihood loss on $X^{(\text{test})}$, $-\sum_{i=1}^{n}\sum_{j=1}^{d} \log f\left(X_{ij}^{(\text{test})} \middle| \epsilon^{(\text{test})} r, p_{ij}^{(K)}\right)$, where $f(\cdot \mid r, p)$ is the density function for the Binomial(r, p) distribution.

We use Algorithm 4 to select the number of clusters in gamma data, as in Example 12, using data thinning.

Algorithm 4: Evaluating gamma clusters with negative log-likelihood loss

Input : A positive integer K, and $X \in \mathbb{R}_{>0}^{n \times d}$ where $X_{ij} \overset{\text{ind.}}{\sim} \text{Gamma}(\lambda, \theta_{c_i,j})$. Here, $\theta \in (0,\infty)^{K^* \times d}$ where $\theta_{c_i,j}$ is the true but unknown rate parameter for the c_ith cluster in the jth dimension, $c_i \in \{1, 2, \ldots, K^*\}$, and λ is the known shape parameter. Also, positive scalars $\epsilon^{(\text{train})}$ and $\epsilon^{(\text{test})} = 1 - \epsilon^{(\text{train})}$.

1 Apply data thinning to X to obtain $X^{(\text{train})}$ and $X^{(\text{test})}$, where $X_{ij}^{(\text{train})} \overset{\text{ind.}}{\sim} \text{Gamma}\left(\epsilon^{(\text{train})}\lambda, \theta_{c_i,j}\right)$ and $X_{ij}^{(\text{test})} \overset{\text{ind.}}{\sim} \text{Gamma}\left(\epsilon^{(\text{test})}\lambda, \theta_{c_i,j}\right)$.

2 Run K-means on $X^{(\text{train})}$ to estimate K clusters. Denote the cluster assignment of the ith observation as $\hat{c}_i$.

3 Within each cluster, estimate the parameters using $X^{(\text{train})}$ (Ye and Chen, 2017; Louzada et al., 2019). Let $\hat{\lambda}^{(K)}$ and $\hat{\theta}^{(K)}$ denote the $K \times d$ estimated parameter matrices.

4 Compute the loss on $X^{(\text{test})}$ as $-\sum_{i=1}^{n}\sum_{j=1}^{d} \log f\left(X_{ij}^{(\text{test})} \middle| \hat{\lambda}_{\hat{c}_i,j}^{(K)} \epsilon^{(\text{test})}/\epsilon^{(\text{train})}, \hat{\theta}_{\hat{c}_i,j}^{(K)}\right)$, where $f(\cdot \mid \lambda, \theta)$ is the density function for the Gamma(λ, θ) distribution.

We apply Algorithms 3 and 4 in three different ways. First, we apply them without modification, with $\epsilon^{(\text{train})} = 0.5$ and $\epsilon^{(\text{train})} = 0.8$. Next, we slightly modify these algorithms by replacing step 1 with multi-fold thinning (Algorithm 2) with $M = 5$ and $\epsilon_1 = \cdots = \epsilon_M = 0.2$. For $m = 1, \ldots, M$, we then perform steps 2–4 using $X^{(\text{train})} = X - X^{(m)}$, $\epsilon^{(\text{train})} = (M-1)/M$ and $X^{(\text{test})} = X^{(m)}$, $\epsilon^{(\text{test})} = 1/M$. We then average the loss functions obtained across the M applications of step 4. Finally, we consider a naive method that re-uses data, by skipping step 1, and simply taking $X^{(\text{train})} = X^{(\text{test})} = X$ in steps 2–4 and $\epsilon^{(\text{train})} = \epsilon^{(\text{test})} = 1$ in step 4.

Our goal is to select the value of K that minimizes the loss function. Because data thinning produces independent training and test sets, we expect that the data thinning approaches will produce U-shaped loss function curves, as a function of K. By contrast, in the naive approach, the full data X is used to fit the model and to compute the loss functions in Algorithms 3 and 4, resulting in monotonically decreasing loss curves, as a function of K.

Other loss functions can be used in lieu of the negative log-likelihood loss in Algorithms 3 and 4. In Appendix E, we extend Algorithms 3 and 4 to the case of mean squared error loss, and show similar results.

6.3 Results

Figure 4 displays the loss function for all three simulation settings as a function of K; results have been averaged over 2,000 simulated data sets and rescaled to the $[0, 1]$ interval for ease of comparison. The values of K with the lowest average loss function are circled on the plots. As expected, the data thinning approaches in Figure 4 exhibit sharp minimum values, as opposed to the monotonically decreasing curves produced by the naive method. The data thinning approaches correctly select the true value of $K = K^*$ in all three settings, except for data thinning with $\epsilon^{(\text{train})} = 0.5$ in the binomial principal components setting. In that case, the low value of $\epsilon^{(\text{train})}$ allocates too much information to the test set, resulting in inadequate signal from the weakest principal components in the training set. Selecting a larger value of $\epsilon^{(\text{train})}$ remedies this issue, as seen with $\epsilon^{(\text{train})} = 0.8$.

We further investigate the role of $\epsilon^{(\text{train})}$ by repeating the simulation study using different values of $\epsilon^{(\text{train})}$ for single-fold data thinning. In Figure 5, we plot the proportion of simulations that select

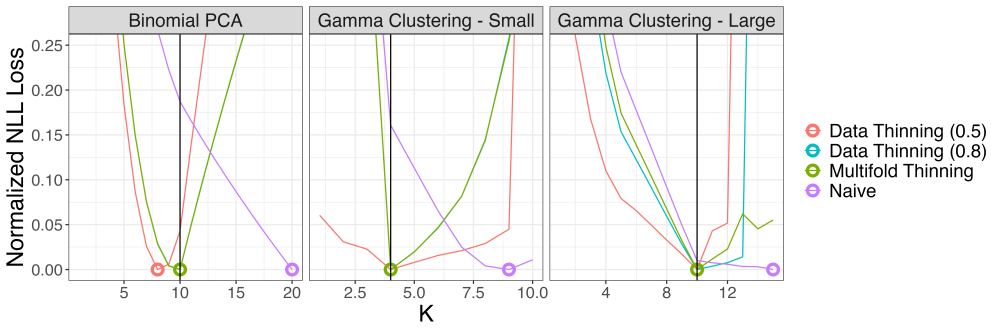

Figure 4: The negative log-likelihood loss averaged over 2,000 simulated data sets, as a function of K, for the naive method (purple), data thinning with $\epsilon^{(\text{train})} = 0.5$ (red), data thinning with $\epsilon^{(\text{train})} = 0.8$ (blue), and multifold thinning with $M = 5$ folds (green). Each curve has been rescaled to take on values between 0 and 1, for ease of comparison. The minimum loss values for each method are circled, and K^* is indicated by the vertical black line.

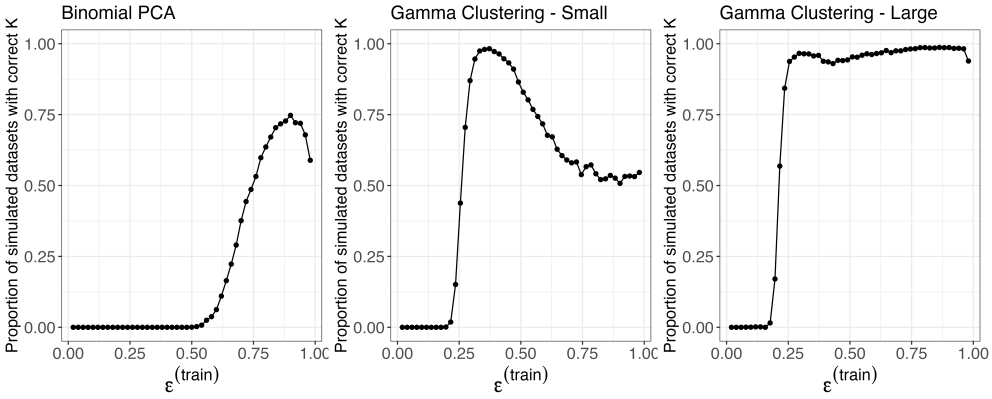

Figure 5: The proportion of simulations for which data thinning selects the true value of K^* with the negative log-likelihood loss, as a function of $\epsilon^{(\text{train})}$, for the simulation study described in Section 6.1. The optimal value of $\epsilon^{(\text{train})}$ depends on the problem at hand.

the correct value of K^* (i.e. the proportion of simulations in which the loss function is minimized at $K = K^*$) in each of the three settings, as a function of $\epsilon^{(\text{train})}$. We find that in the gamma clustering simulations, lower values of $\epsilon^{(\text{train})}$ are adequate. However, settings with weaker signal, such as the binomial principal components example, require larger values of $\epsilon^{(\text{train})}$ to identify the true latent structure. In all settings, as $\epsilon^{(\text{train})}$ approaches 1, performance begins to decay. This is a consequence of inadequate information remaining in the test set under large values of $\epsilon^{(\text{train})}$, and is consistent with the discussion of Section 4.1. These findings suggest that in practice, the optimal value of $\epsilon^{(\text{train})}$ is context-dependent.

Finally, we examine the benefits of multifold data thinning over single-fold data thinning. Figure 6 displays histograms of the number of simulations that select each value of K. Here we only include data thinning with $\epsilon^{(\text{train})} = 0.8$ and multifold thinning with $M = 5$, so that both methods use the same allocation of information between training and test sets. We see that multifold thinning generally selects the correct value of K more often than single-fold data thinning, mirroring the

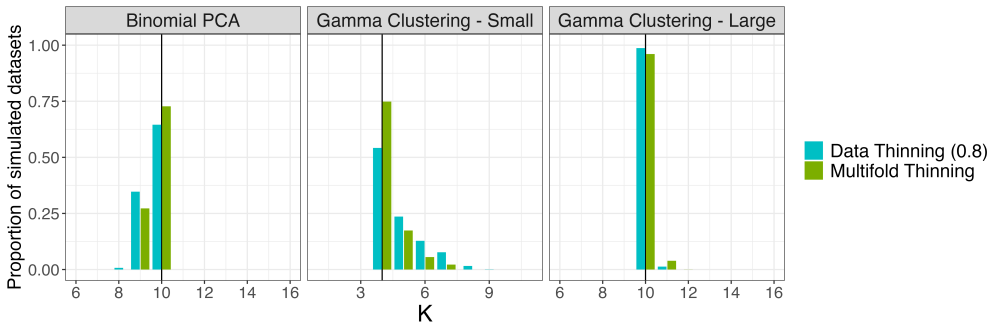

Figure 6: The proportion of simulated data sets in which each candidate value of K is selected, with the negative log-likelihood loss, under data thinning with $\epsilon^{(\text{train})} = 0.8$ (blue) and multifold thinning with $M = 5$ (green), for each of the simulation settings described in Section 6.1. The true value of K^* is indicated by the vertical black line. Multifold thinning tends to select the true value of K more often than single-fold thinning.

improvement of M-fold cross-validation using sample splitting over single-fold sample splitting in supervised settings. However, in the large gamma setting, the signal is strong enough that multifold thinning does not provide a benefit over single-fold thinning.

7. Selecting the Number of Principal Components in Gene Expression Data

In this section, we revisit an analysis of a data set from a single-cell RNA sequencing experiment conducted on a set of peripheral blood mononuclear cells. The data set is freely available from 10X Genomics, and was previously analyzed in the "Guided Clustering Tutorial" vignette (Hoffman et al., 2022) for the popular `R` package `Seurat` (Hao et al., 2021; Stuart et al., 2019; Satija et al., 2015).

The data set X is a sparse matrix of non-negative integers, representing counts from $32,738$ genes in each of $2,700$ cells. We consider applying principal components analysis to learn a low-dimensional representation of the data. In the `Seurat` vignette, filtering, normalization, log-transformation, feature selection, centering, and scaling are applied to the data, yielding a transformed matrix $\tilde{Y} \in \mathbb{R}^{2638 \times 2000}$. Details are provided in Section F.1. Finally, the singular value decomposition of $\tilde{Y}$ is computed, such that $\tilde{Y} = UDV^T$. Here we let U_k represent the kth column of the matrix U, and let $U_{1:K}D_{1:K}V_{1:K}^T$ represent the rank-K approximation of $\tilde{Y}$.

Our goal is to select the number of dimensions to use in this low-rank approximation. In the `Seurat` vignette, the authors rely on heuristic solutions such as looking for an elbow in the plot of the standard deviation of $U_K D_K$ as a function of K; see Figure 7(a) (James et al., 2013). Based on the elbow plot, the authors suggest retaining around 7 principal components. Other heuristic approaches suggest as many as 12 principal components.

Before introducing the data thinning solution, we introduce a squared-error based formulation that is mathematically equivalent to the traditional elbow plot (see Section F.2), but will facilitate a direct comparison with data thinning. For $K = 1, \ldots, 20$, we compute the sum of squared errors between the matrix $\tilde{Y}$ and its rank-K approximation:

$$\left\| \tilde{Y} - U_{1:K} D_{1:K} V_{1:K}^T \right\|_F^2.$$

Because the low-rank approximation $U_{1:K}D_{1:K}V_{1:K}^T$ is computed using $\tilde{Y}$, this loss function monotonically decreases with K. A heuristic solution for deciding how many principal components to retain involves looking for the point in which the slope of the curve in Figure 7(b) begins to flatten. While this appears to happen around 5–7 principal components, which is consistent with the finding from Figure 7(a), the exact number of principal components to retain is still unclear. We now show that data thinning provides a principled approach for estimating the number of principal components.

Single-cell RNA-sequencing data are often modeled as independent Poisson random variables (Wang et al., 2018; Sarkar and Stephens, 2021). Thus, we assume that $X_{ij} \sim \text{Poisson}(\Lambda_{ij})$. Starting with the raw data matrix $X \in \mathbb{Z}_{\geq 0}^{2700 \times 32738}$, we perform Poisson data thinning with $\epsilon = 0.5$ to obtain a training set $X^{(1)}$ and a test set $X^{(2)}$, which are independent if the Poisson assumption holds. Furthermore, as $\epsilon = 0.5$, they are identically distributed. We then carry out the data processing described in Section F.1 on $X^{(1)}$ to obtain $\tilde{Y}^{(1)} \in \mathbb{R}^{2638 \times 2000}$. We obtain $\tilde{Y}^{(2)} \in \mathbb{R}^{2638 \times 2000}$ by applying the same data processing steps to $X^{(2)}$, but retaining only the features that were selected on $X^{(1)}$, so that the rows and columns of $\tilde{Y}^{(1)}$ and $\tilde{Y}^{(2)}$ correspond to the same genes and cells. Details are in Section F.1.

We compute the singular value decomposition on the training set, $\tilde{Y}^{(1)} = U^{(1)}D^{(1)}(V^{(1)})^T$. For a range of values of K, we then compute the sum of squared errors between $\tilde{Y}^{(2)}$ and $U_{1:K}^{(1)}D_{1:K}^{(1)}(V_{1:K}^{(1)})^T$:

$$\left\| \tilde{Y}^{(2)} - U_{1:K}^{(1)} D_{1:K}^{(1)} (V_{1:K}^{(1)})^T \right\|_F^2. \tag{1}$$

The results are shown in Figure 7(c). As we are not computing and evaluating the singular value decomposition using the same data, the plot of K vs. the loss function is not monotonically decreasing in K. Instead, it reaches a clear minimum at $K = 7$, suggesting that the rank-7 approximation provides the best fit to the observed data.

While our analysis thus far assumes a Poisson model for single-cell RNA sequencing data, there is evidence that a negative binomial model may be preferable in some settings (Choudhary and Satija, 2022). Thus, we modified our analysis to use negative binomial thinning instead of Poisson thinning. Details are provided in Appendix F.3, and the results are shown in Figure 7(d). This modified analysis yields similar results: it suggests that we retain $K = 5$ principal components.

There are two potential reasons for the discrepancy between $K = 5$ and $K = 7$.

1. Suppose that we perform data thinning under the Poisson assumption, but the data are in fact overdispersed relative to the Poisson distribution. Then there will be positive correlation between $X^{(1)}$ and $X^{(2)}$. This could cause Poisson thinning to overestimate the number of principal components to retain in the low-rank approximation of X.

2. The negative binomial thinning procedure injects more noise into $X^{(1)}$ than does the Poisson thinning procedure. This could cause negative binomial thinning to miss out on true signal in the data, and thus underestimate the number of principal components.

While either of these explanations is plausible, it is comforting to see that the two methods provide similar estimates for K. The take home message is that, for a given distributional assumption, data thinning provides a simple and non-heuristic way to select the number of principal components.

Remark 18 (Choice of ϵ) *If we had chosen $\epsilon \neq 0.5$, then while $E[X^{(2)}] = \frac{1-\epsilon}{\epsilon} E[X^{(1)}]$, the relationship between $E[\tilde{Y}^{(1)}]$ and $E[\tilde{Y}^{(2)}]$ would depend on the details of the data processing described in Section 2, and the loss function in* (1) *would need to be modified accordingly.*

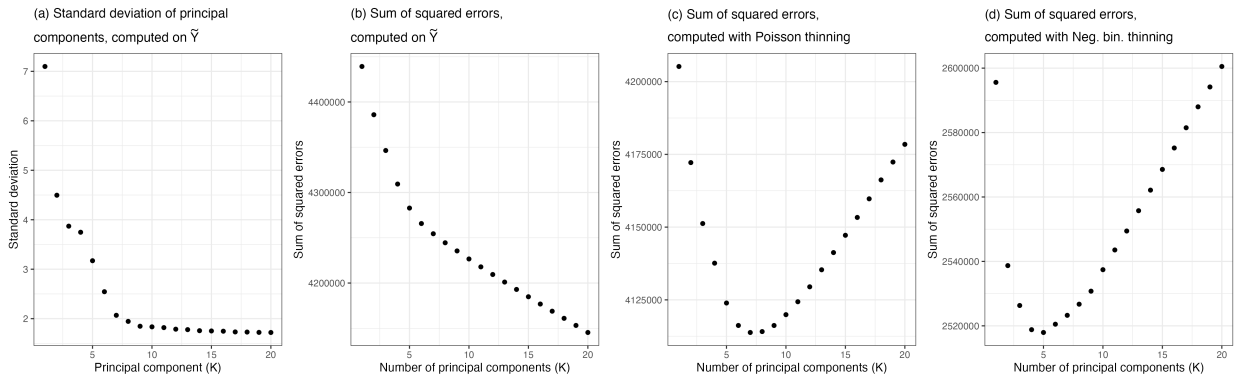

Figure 7: Results for the data analysis in Section 7. *(a)* An "elbow plot" of the standard deviation of the principal components, which reproduces the plot given in the Seurat guided clustering tutorial. *(b)* Due to the relationship between the sum of squared errors and the standard deviation of the principal components (see Section F.2), looking for an elbow in (a) is equivalent to looking for an elbow in (b). *(c)* We perform Poisson data thinning to obtain a training set (used to compute the singular value decompoistion) and a test set (used to compute the sum of squared errors). The sum of squared errors is minimized at 7 principal components. *(d)* The negative binomial data thinning version of (c), which shows a minimum in the loss function at 5 principal components.

8. Discussion

We have proposed data thinning, a new way to decompose a single observation into two or more independent observations that sum to yield the original. This proposal applies to a very broad class of distributions. Furthermore, we have compared data thinning to sample splitting, and have shown that the former is applicable in many cases that the latter is not, and may be preferable even when both are applicable.

In Section 2.3, we considered the impact of using the incorrect value of a nuisance parameter when performing data thinning, but we did not consider what happens when the nuisance parameter is estimated using the data itself. In future work, we will consider the theoretical and empirical implications of performing data thinning with an estimated nuisance parameter. Furthermore, we focused on convolution-closed distributions and thus used additive decompositions where $X = X^{(1)} + X^{(2)}$. For distributions with bounded support, such as the beta distribution, non-additive decompositions are needed. We leave such decompositions to future work.

An R package implementing data thinning and scripts to reproduce the results in this paper are available at `https://anna-neufeld.github.io/datathin/`.

Acknowledgments

This material is based upon research supported in part by the Office of Naval Research (award number N000142312589), the National Science Foundation (award Number 2322920), the Simons Foundation (Simons Investigator Award in Mathematical Modeling of Living Systems), the National Institutes of Health (R01 EB026908, R01 GM123993, and R01 DA047869), and the Keck

Foundation. Lucy Gao and Ameer Dharamshi were supported in part by the Natural Sciences and Engineering Research Council of Canada.

Appendix A. Proofs from Section 2

A.1 Proof of Theorem 5

Proof This proof is due to Joe (1996) and Jørgensen and Song (1998), but has been adapted to fit our notation. Let x, our observed data, be a realization of a random variable $X \sim F_\lambda$. Let $\epsilon \in (0,1)$ be chosen such that $\epsilon\lambda$ and $(1-\epsilon)\lambda$ are in the parameter space Λ. We draw $X^{(1)} \mid X = x \sim G_{\epsilon\lambda,(1-\epsilon)\lambda,x}$, where this notation was defined in Section 2.1, and let $X^{(2)} = X - X^{(1)}$.

Separately, let $X' \sim F_{\epsilon\lambda}$ and $X'' \sim F_{(1-\epsilon)\lambda}$ be independent, and let $Y = X' + X''$. As F_λ is a convolution-closed distribution, it follows that $Y \sim F_\lambda$ and thus we know that Y has the same marginal distribution as X.

We first argue that the joint distribution of $(X^{(1)}, X)$ is the same as the joint distribution of (X', Y). The conditional distribution of $X^{(1)} \mid X$ is the same as the conditional distribution of $X' \mid Y$ by definition of the distribution $G_{\epsilon\lambda,(1-\epsilon)\lambda,x}$. Furthermore, we have already noted that the marginal distributions of X and Y are the same. Thus, the joint distribution of $(X^{(1)}, X)$ is the same as the joint distribution of (X', Y).

As $X^{(2)}$ is deterministic given $X^{(1)}$ and X, the joint distribution of $(X^{(1)}, X^{(2)})$ is the same as the joint distribution of $(X^{(1)}, X)$. Similarly, the joint distribution of (X', Y) is the same as the joint distribution of (X', X''). Thus, the joint distribution of $(X^{(1)}, X^{(2)})$ is the same as the joint distribution of (X', X''). As the joint distribution of X' and X'' is known to be the product of independent distributions $F_{\epsilon\lambda}$ and $F_{(1-\epsilon)\lambda}$, this completes the proof of parts (i) and (ii) of Theorem 5.

The final statement of Theorem 5 follows directly from Definition 2. ■

A.2 Proof of Proposition 10

Proof To prove (i), note that since $X^{(1)} \mid X = x$ is normally distributed and X is normally distributed, a well-known property of the normal distribution tells us that the marginal distribution of $X^{(1)}$ is normal. We then use the law of total expectation and the law of total variance to compute its mean and variance.

$$\begin{aligned}
\mathrm{E}[X^{(1)}] &= \mathrm{E}[\mathrm{E}[X^{(1)} \mid X]] = \mathrm{E}[\epsilon X] = \epsilon\mu \\
\mathrm{Var}(X^{(1)}) &= \mathrm{Var}\left(\mathrm{E}\left[X^{(1)} \mid X\right]\right) + \mathrm{E}\left[\mathrm{Var}\left(X^{(1)} \mid X\right)\right] \\
&= \mathrm{Var}\left(\epsilon X\right) + \mathrm{E}\left(\epsilon(1-\epsilon)\tilde{\sigma}^2\right) \\
&= \epsilon^2\sigma^2 + \epsilon(1-\epsilon)\tilde{\sigma}^2.
\end{aligned}$$

To prove (ii), note that the difference between two normally distributed variables (X and $X^{(1)}$) is normal. Then note that

$$\begin{aligned}
\mathrm{E}[X^{(2)}] &= \mathrm{E}[X] - \mathrm{E}[X^{(1)}] = \mu - \epsilon\mu = (1-\epsilon)\mu. \\
\mathrm{Var}(X^{(2)}) &= \mathrm{Var}\left(\mathrm{E}\left[X^{(2)} \mid X\right]\right) + \mathrm{E}\left[\mathrm{Var}\left(X^{(2)} \mid X\right)\right] \\
&= \mathrm{Var}\left(\mathrm{E}\left[X - X^{(1)} \mid X\right]\right) + \mathrm{E}\left[\mathrm{Var}\left(X - X^{(1)} \mid X\right)\right] \\
&= \mathrm{Var}\left((1-\epsilon)X\right) + \mathrm{E}\left[\mathrm{Var}\left(X^{(1)} \mid X\right)\right] \\
&= (1-\epsilon)^2\sigma^2 + \epsilon(1-\epsilon)\tilde{\sigma}^2,
\end{aligned}$$

which completes the proof of (ii). Finally, to prove (iii), note that

$$\begin{aligned}
2\operatorname{Cov}(X^{(1)}, X^{(2)}) &= \operatorname{Var}(X) - \operatorname{Var}(X^{(1)}) - \operatorname{Var}(X^{(2)}) \\
&= \sigma^2 - \epsilon^2\sigma^2 - \epsilon(1-\epsilon)\tilde{\sigma}^2 - (1-\epsilon)^2\sigma^2 - \epsilon(1-\epsilon)\tilde{\sigma}^2 \\
&= 2\epsilon(1-\epsilon)\left(\sigma^2 - \tilde{\sigma}^2\right).
\end{aligned}$$

■

A.3 Proof of Proposition 11

Proof Recall that if $A \sim \text{BetaBinomial}\,(r, \alpha, \beta)$, then $\mathrm{E}[A] = \frac{r\alpha}{\alpha+\beta}$ and $\operatorname{Var}(A) = \frac{r\alpha\beta(\alpha+\beta+r)}{(\alpha+\beta)^2(\alpha+\beta+1)}$. Then we can derive the marginal variance of $X^{(1)}$ using the law of total variance and the fact that $X^{(1)} \mid X \sim \text{BetaBinomial}(X, \epsilon\tilde{r}, (1-\epsilon)\tilde{r})$.

$$\begin{aligned}
\operatorname{Var}(X^{(1)}) &= \mathrm{E}[\operatorname{Var}(X^{(1)} \mid X)] + \operatorname{Var}(\mathrm{E}[X^{(1)} \mid X]) \\
&= \mathrm{E}\left[\frac{X\epsilon(1-\epsilon)(\tilde{r}+X)}{(\tilde{r}+1)}\right] + \operatorname{Var}\left(\epsilon X\right) \\
&= \frac{\epsilon(1-\epsilon)}{\tilde{r}+1}\left(\tilde{r}\,\mathrm{E}\left[X\right] + \mathrm{E}\left[X^2\right]\right) + \epsilon^2\operatorname{Var}\left(X\right) \\
&= \frac{\epsilon(1-\epsilon)}{\tilde{r}+1}\left(\tilde{r}\,\mathrm{E}\left[X\right] + \operatorname{Var}(X) + E[X]^2\right) + \epsilon^2\operatorname{Var}\left(X\right).
\end{aligned}$$

Next note that $\operatorname{Var}(X^{(2)} \mid X) = \operatorname{Var}(X - X^{(1)} \mid X) = \operatorname{Var}(X^{(1)} \mid X)$ and that $\mathrm{E}[X^{(2)} \mid X] = (1-\epsilon)X$. Thus, we arrive at:

$$\operatorname{Var}(X^{(2)}) = \frac{\epsilon(1-\epsilon)}{\tilde{r}+1}\left(\tilde{r}\,\mathrm{E}\left[X\right] + \operatorname{Var}(X) + E[X]^2\right) + (1-\epsilon)^2\operatorname{Var}\left(X\right).$$

To derive the covariance, note that:

$$\begin{aligned}
2\operatorname{Cov}(X^{(1)}, X^{(2)}) &= \operatorname{Var}(X) - \operatorname{Var}(X^{(1)}) - \operatorname{Var}(X^{(2)}) \\
&= \operatorname{Var}(X) - 2\frac{\epsilon(1-\epsilon)}{\tilde{r}+1}\left(\tilde{r}\,\mathrm{E}\left[X\right] + \operatorname{Var}(X) + E[X]^2\right) - \left(\epsilon^2 + (1-\epsilon)^2\right)\operatorname{Var}\left(X\right) \\
&= -2\frac{\epsilon(1-\epsilon)}{\tilde{r}+1}\left(\tilde{r}r\frac{1-p}{p} + r\frac{1-p}{p^2} + r^2\frac{(1-p)^2}{p^2}\right) + 2\epsilon(1-\epsilon)r\frac{1-p}{p^2} \\
&= 2\epsilon(1-\epsilon)r\frac{(1-p)^2}{p^2}\left(1 - \frac{r+1}{\tilde{r}+1}\right).
\end{aligned}$$

■

A.4 Proof of Proposition 12

Proof The proof structure is identical to those of Proposition 10 and Proposition 11. We start by recalling that if $A \sim \text{Gamma}(\alpha, \beta)$, then $\text{E}[X] = \frac{\alpha}{\beta}$ and $\text{Var}(X) = \frac{\alpha}{\beta^2}$. Then:

$$\begin{aligned}
\text{Var}(X^{(1)}) &= E\left[\text{Var}\left(X^{(1)} \mid X\right)\right] + \text{Var}\left(\text{E}\left[X^{(1)} \mid X\right]\right) \\
&= E\left[\text{Var}\left(XZ \mid X\right)\right] + \text{Var}\left(\text{E}\left[XZ \mid X\right]\right) \\
&= E\left[X^2 \,\text{Var}\left(Z\right)\right] + \text{Var}\left(X \,\text{E}\left[Z\right]\right) \\
&= E\left[X^2 \left(\frac{\epsilon(1-\epsilon)}{\tilde{\alpha}+1}\right)\right] + \text{Var}\left(X\epsilon\right) \\
&= \frac{\epsilon(1-\epsilon)}{\tilde{\alpha}+1}\left(\text{Var}(X) + \text{E}[X]^2\right) + \epsilon^2 \,\text{Var}\left(X\right).
\end{aligned}$$

Similarly, we note that $\text{Var}(X^{(2)} \mid X) = \text{Var}(X - X^{(1)} \mid X) = \text{Var}(X^{(1)} \mid X)$, while $\text{E}(X^{(2)} \mid X) = (1-\epsilon)X$. This allows us to do a similar derivation and arrive at:

$$\begin{aligned}
\text{Var}(X^{(2)}) &= E\left[\text{Var}\left(X^{(2)} \mid X\right)\right] + \text{Var}\left(\text{E}\left[X^{(2)} \mid X\right]\right) \\
&= \frac{\epsilon(1-\epsilon)}{\tilde{\alpha}+1}\left(\text{Var}(X) + \text{E}[X]^2\right) + (1-\epsilon)^2 \,\text{Var}\left(X\right).
\end{aligned}$$

Finally,

$$\begin{aligned}
2\,\text{Cov}(X^{(1)}, X^{(2)}) &= \text{Var}(X) - \text{Var}(X^{(1)}) - \text{Var}(X^{(2)}) \\
&= \text{Var}(X) - 2\frac{\epsilon(1-\epsilon)}{\tilde{\alpha}+1}\left(\text{Var}(X) + \text{E}[X]^2\right) - \left(\epsilon^2 + 1 - 2\epsilon + \epsilon^2\right)\text{Var}(X) \\
&= -2\frac{\epsilon(1-\epsilon)}{\tilde{\alpha}+1}\left(\text{Var}(X) + \text{E}[X]^2\right) + 2\epsilon(1-\epsilon)\,\text{Var}(X) \\
&= -2\frac{\epsilon(1-\epsilon)}{\tilde{\alpha}+1}\left(\frac{\alpha(1+\alpha)}{\beta^2}\right) + 2\epsilon(1-\epsilon)\frac{\alpha}{\beta^2} \\
&= 2\epsilon(1-\epsilon)\frac{\alpha}{\beta^2}\left(1 - \frac{\alpha+1}{\tilde{\alpha}+1}\right).
\end{aligned}$$

■

Appendix B. Proof of Theorem 13

Proof The proof is nearly identical to that of Theorem 5. It extends ideas from Jørgensen and Song (1998) and Joe (1996) to the setting of multiple folds.

Let x, our observed data, be a realization of random variable $X \sim F_\lambda$. Let $\epsilon_1, \ldots, \epsilon_M$ be chosen such that $\sum_{m=1}^M \epsilon_m = 1$, $\epsilon_m > 0$, and $\epsilon_m \lambda$ is in the parameter space Λ for $m = 1, \ldots, M$.

Suppose we draw $\left(X^{(1)}, \ldots, X^{(M)}\right) \mid X = x \sim G_{\epsilon_1\lambda, \epsilon_2\lambda, \ldots, \epsilon_M\lambda, x}$, where $G_{\epsilon_1\lambda, \epsilon_2\lambda, \ldots, \epsilon_M\lambda, x}$ was defined in Section 3.

Separately, let $X_1, X_2, \ldots, X_M$ be mutually independent random variables, where $X_m \sim F_{\epsilon_m\lambda}$, and let $Y = \sum_{m=1}^M X_m$. As F_λ is a convolution-closed distribution, we know that $Y \sim F_\lambda$ and thus Y has the same marginal distribution as X.

The conditional joint distribution of $(X^{(1)}, X^{(2)}, \ldots, X^{(M)}) \mid X$ is the same as the conditional joint distribution of $(X_1, X_2, \ldots, X_M) \mid Y$, by definition of the distribution $G_{\epsilon_1\lambda,\ldots,\epsilon_M\lambda,x}$. Furthermore, we have already seen that the marginal distributions of X and Y are the same. Thus, the marginal joint distribution of $(X^{(1)}, X^{(2)}, \ldots, X^{(M)})$ is the same as the marginal joint distribution of $(X_1, X_2, \ldots, X_M)$. Furthermore, by construction, the marginal joint distribution of $(X_1, X_2, \ldots, X_M)$ is that of M mutually independent random variables, where $X_m \sim F_{\epsilon_m\lambda}$. This concludes the proof of Theorem 13, parts 1-3. Part 4 of Theorem 13 follows directly from Definition 2. ■

Appendix C. Proof of Theorem 14

To prove Theorem 14, we rely on Lemma 19 and Lemma 20.

Lemma 19 *Suppose that we thin a random variable $X \sim F_\lambda$ using Algorithm 2 with $\epsilon_1 = \ldots = \epsilon_M = \frac{1}{M}$ to obtain $X^{(1)}, \ldots, X^{(M)}$. Let $I_X(\theta)$ denote the Fisher information contained in X about an unknown parameter θ (assume that this Fisher information exists). Then the Fisher information contained in $X^{(m)}$ for $m = 1, \ldots, M$ about θ, denoted $I_{X^{(m)}}(\theta)$, is equal to $\frac{1}{M} I_X(\theta)$.*

Proof We began with a random variable X and we constructed $\left(X^{(1)}, \ldots, X^{(m)}\right)$ without knowledge of θ. We cannot create information from nothing. Thus, if $I_{\left(X^{(1)},\ldots,X^{(m)}\right)}(\theta)$ denotes the information about θ in the joint distribution of $\left(X^{(1)}, \ldots, X^{(m)}\right)$, then

$$I_{\left(X^{(1)},\ldots,X^{(m)}\right)}(\theta) \leq I_X(\theta). \tag{2}$$

To prove (2) formally, we can use the chain rule property of Fisher information to write $I_{\left(X^{(1)},\ldots,X^{(m)},X\right)}(\theta)$ in two ways:

$$I_{\left(X^{(1)},\ldots,X^{(m)},X\right)}(\theta) = I_{\left(X^{(1)},\ldots,X^{(m)}\right)}(\theta) + I_{X|\left(X^{(1)},\ldots,X^{(m)}\right)}(\theta) = I_{\left(X^{(1)},\ldots,X^{(m)}\right)|X}(\theta) + I_X(\theta).$$

Since θ is unknown during the thinning process, the distribution $\left(X^{(1)}, \ldots, X^{(m)}\right) \mid X$ must not involve θ, so $I_{\left(X^{(1)},\ldots,X^{(m)}\right)|X}(\theta) = 0$. This implies that

$$I_{\left(X^{(1)},\ldots,X^{(m)}\right)}(\theta) + I_{X|\left(X^{(1)},\ldots,X^{(m)}\right)}(\theta) = I_X(\theta),$$

and since Fisher information is non-negative, (2) follows.

Separately, since $X = X^{(1)} + \ldots + X^{(m)}$, we can reconstruct X from $\left(X^{(1)}, \ldots, X^{(m)}\right)$, and so a fundamental property of Fisher information tells us that

$$I_X(\theta) = I_{\sum_{m=1}^{M} X^{(m)}}(\theta) \leq I_{\left(X^{(1)},\ldots,X^{(m)}\right)}(\theta). \tag{3}$$

Combining (2) and (3), we have that

$$I_X(\theta) = I_{\left(X^{(1)},\ldots,X^{(m)}\right)}(\theta).$$

Finally, as $X^{(1)}, \ldots, X^{(M)}$ are independent and identically distributed,

$$I_X(\theta) = I_{\left(X^{(1)},\ldots,X^{(M)}\right)}(\theta) = \sum_{m=1}^{M} I_{X^{(m)}}(\theta) = M I_{X^{(1)}}(\theta).$$

It follows immediately that $I_{X^{(1)}}(\theta) = \frac{1}{M} I_X(\theta)$. Similarly, $I_{X^{(m)}}(\theta) = \frac{1}{M} I_X(\theta)$ for $m = 2, \ldots, M$. ■

Lemma 20 *In the setting of Lemma 19, the Fisher information contained in* $\sum_{m=1}^{K} X^{(m)}$ *for* $K < M$*, denoted* $I_{\sum_{m=1}^{K} X^{(m)}}(\theta)$*, is equal to* $\frac{K}{M} I_X(\theta)$*.*

Proof The convolution-closed property of F_λ says that $\sum_{m=1}^{K} X^{(m)} \sim F_{\frac{K}{M}\lambda}$. As $\sum_{m=1}^{K} X^{(m)}$ is a function of $X^{(1)}, \ldots, X^{(K)}$, we have that

$$I_{\left(X^{(1)}, \ldots, X^{(K)}\right)}(\theta) \geq I_{\sum_{m=1}^{K} X^{(m)}}(\theta).$$

We can construct a random variable $Y \sim F_{\frac{K}{M}\lambda}$ with the same distribution as $\sum_{m=1}^{K} X^{(m)}$ by applying Algorithm 2 to X with $\epsilon_1 = K/M$ and $\epsilon_2 = 1 - K/M$ and calling the first fold of data Y. We then can thin Y with $\epsilon_1 = \ldots = \epsilon_K = \frac{1}{K}$ to obtain $Y^{(1)}, \ldots, Y^{(K)}$, whose joint and marginal distributions are equal to $X^{(1)}, \ldots, X^{(K)}$ (independent random variables that follow $F_{\frac{1}{M}\lambda}$). This allows us to rewrite the inequality above as

$$I_{\left(Y^{(1)}, \ldots, Y^{(K)}\right)}(\theta) = I_{\left(X^{(1)}, \ldots, X^{(K)}\right)}(\theta) \geq I_{\sum_{m=1}^{K} X^{(m)}}(\theta) = I_Y(\theta). \tag{4}$$

By the same logic that was given in the proof of Lemma 19, we know that we can produce $I_{\left(Y^{(1)}, \ldots, Y^{(K)}\right)}$ from Y without knowing θ. Thus, $Y^{(1)}, \ldots, Y^{(K)}$ cannot contain more information about θ than Y. Thus, we have that

$$I_{\left(Y^{(1)}, \ldots, Y^{(K)}\right)}(\theta) = I_{\left(X^{(1)}, \ldots, X^{(K)}\right)}(\theta) \leq I_{\sum_{m=1}^{K} X^{(m)}}(\theta) = I_Y(\theta), \tag{5}$$

where the last equality holds since two random variables with the same distribution contain the same amount of information about θ. Combining the inequalities in (4) and (5) yields

$$I_{\sum_{m=1}^{K} X^{(m)}}(\theta) = I_{\left(X^{(1)}, \ldots, X^{(K)}\right)}(\theta).$$

Finally, we note that

$$I_{\sum_{m=1}^{K} X^{(m)}}(\theta) = I_{\left(X^{(1)}, \ldots, X^{(K)}\right)}(\theta) = \sum_{m=1}^{K} I_{X^{(m)}}(\theta) = \frac{K}{M} I_X(\theta),$$

where the second equality follows from independence, and the third from Lemma 19. This completes the proof. ■

We now proceed with the proof of Theorem 14.

Proof For simplicity, assume that ϵ_m is a rational number for $m = 1, \ldots, M$. Then, we can rewrite $\epsilon_1, \ldots, \epsilon_M$ as $\frac{K_1}{M^*}, \frac{K_2}{M^*}, \ldots, \frac{K_M}{M^*}$ for integers $M^*, K_1, \ldots, K_M$. Further assume that $\frac{1}{M^*}\lambda$ is in the parameter space Λ for our model. Then, for $m = 1, \ldots, M$, $X^{(m)}$ has the same distribution as $\sum_{j=1}^{K_m} \tilde{X}^{(j)}$, where $\tilde{X}^{(1)}, \ldots, \tilde{X}^{(M^*)}$ are random variables that we would obtain if we thinned X into M^* equally sized folds. By Lemma 20, $I_{X^{(m)}}(\theta) = I_{\sum_{j=1}^{K_m} \tilde{X}^{(j)}}(\theta) = \frac{K_m}{M^*} I_X(\theta) = \epsilon_m I_X(\theta)$. ■

Remark 21 *The proof of Theorem 14 assumes that (i) ϵ_m is rational for $m = 1, \ldots, M$, and that (ii) $\frac{1}{M^*}\lambda \in \Lambda$ for the common denominator M^*. Among the distributions in Table 2, only the* Binomial(r, p) *and the* Multinomial(r, p) *have relevant restrictions on the parameter space Λ. To be able to thin one of these distributions with parameter ϵ_m, we must have that $\epsilon_m r$ is a positive integer. Thus, $\epsilon_m = \frac{Z_m}{r}$ for some integer Z_m, i.e. (i) is satisfied. Adopting the notation of the proof of Theorem 14, we note that $K_m = Z_m$ and that $M^* = r = \lambda$. Thus, the requirement that $\frac{\lambda}{M^*}$ is in the parameter space (i.e. that it is a positive integer) follows immediately, since $\frac{\lambda}{M^*} = 1$.*

Appendix D. Simulation Study Supporting Details

In this section, we provide additional details about the simulation studies described in Section 6.1.

For Example 11, in which we select the number of principal components for binomial data, we use the following setup. For $K^* = 10$, we compute $\theta = UDV^T$ where U is a $n \times K^*$ random orthogonal matrix, D is a $K^* \times K^*$ diagonal matrix with diagonal elements equal to $5, 6, \ldots, 14$, and V is a $d \times K^*$ random orthogonal matrix. Then, $p_{ij} = \frac{\exp(\theta_{ij})}{1+\exp(\theta_{ij})}$ for $i = 1, \ldots, n$ and $j = 1, \ldots, d$.

For Example 12, in which we select the number of clusters in gamma-distributed data, we use the following setup.

In the small d, small K^* clustering setting described in Example 12, observations from each cluster are generated as $X_{ij} \overset{\text{ind}}{\sim} \text{Gamma}(\lambda, \theta_{c_i,j})$ where $\lambda = 20$,

$$\theta = \begin{bmatrix} 0.5 & 5 \\ 5 & 0.5 \\ 10 & 10 \\ 0.5 & 0.5 \end{bmatrix},$$

and $c_i \in \{1, 2, 3, 4\}$ is the true cluster membership for the ith observation.

In the large d, large K^* clustering setting described in Example 12, observations from each cluster are generated as $X_{ij} \overset{\text{ind}}{\sim} \text{Gamma}(\lambda, \theta_{c_i,j})$ where $\lambda = 2$, the $K^* \times d$ matrix θ is constructed such that for $j = 1, \ldots, d$ and $k = 1, \ldots, K^*$,

$$\theta_{kj} = \begin{cases} 0.1 & \text{if } k \leq 9 \text{ and } 10k - 9 \leq j \leq 10k + 10, \\ 1 & \text{otherwise,} \end{cases}$$

and $c_i \in \{1, 2, \ldots, 10\}$ is the true cluster membership for the ith observation.

Appendix E. Simulation with Mean Squared Error Loss Function

E.1 Methods

As an alternative to the negative log-likelihood loss used in Section 6, here we consider applying a mean squared error loss. To do this, we simply replace the negative log likelihood from Step 4 of Algorithms 3 and 4 with the mean squared error, defined as

$$\frac{1}{nd}\sum_{i=1}^{n}\sum_{j=1}^{d}\left(X_{ij}^{(\text{test})} - \epsilon^{(\text{test})} r p_{ij}^{(K)}\right)^2 \tag{6}$$

in the case of Algorithm 3 and

$$\frac{1}{nd}\sum_{i=1}^{n}\sum_{j=1}^{d}\left(X_{ij}^{(\text{test})} - \frac{\epsilon^{(\text{test})}}{\epsilon^{(\text{train})}}\hat{\mu}_{\hat{c}_i,j}^{(\text{K})}\right)^2 \tag{7}$$

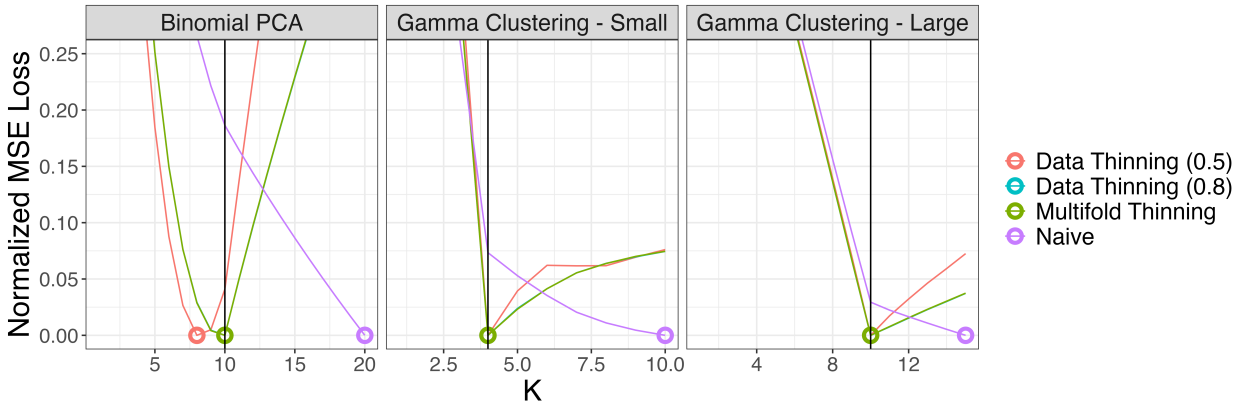

Figure 8: The mean squared error loss averaged over 2,000 simulated data sets, as a function of K, for the naive method (purple), data thinning with $\epsilon^{(\text{train})} = 0.5$ (red), data thinning with $\epsilon^{(\text{train})} = 0.8$ (blue), and multifold thinning with 5 folds (green). Each curve has been rescaled to take on values between 0 and 1, for ease of comparison. The minimum loss values for each method are circled, and K^* is indicated by the vertical black line.

in the case of Algorithm 4.

After replacing the loss functions, these algorithms can be applied directly to obtain simulation results for data thinning, and with slight modification to obtain results for multi-fold data thinning and the naive method, as described in Section 4.2.

E.2 Results

In Figure 8, we plot the average mean squared error curves, as a function of K. As with the negative log-likelihood loss, data thinning approaches produce curves with sharp minimum values at or near $K = K^*$, as opposed to the naive method's monotonically-decreasing curves.

In Figure 9, we plot the proportion of simulations that select the correct value of K^* using the mean squared error loss, as a function of $\epsilon^{(\text{train})}$. Results are largely similar to Figure 5.

Finally, we compare multi-fold to single-fold thinning, under the mean squared error loss, in Figure 10. As in Figure 6, we find that multi-fold thinning tends to select the correct value of K more often than single-fold thinning.

Appendix F. Details for the Real Data Analysis in Section 7

F.1 Preprocessing

We first explain in detail the preprocessing done to the matrix X in the Seurat tutorial.

(1) Initial data filtering: We initially filter the data such that only cells with between 200 and 2500 total counts remain (with fewer than 5% of the counts coming from from mitochondrial genes) and only genes that are expressed in at least 200 cells remain. This reduces the size of X from $2,700 \times 32,738$ to $2,638 \times 13,714$

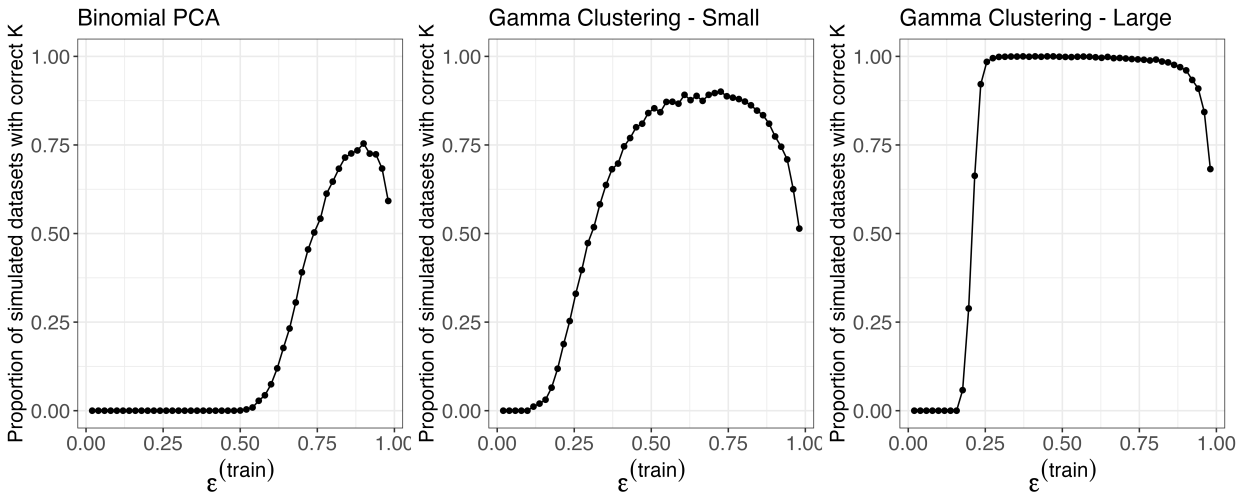

Figure 9: The proportion of simulations for which data thinning selects the true value of K^* with the mean squared error loss, as a function of $\epsilon^{(\text{train})}$, for the simulation study described in Section 6.1. The optimal value of $\epsilon^{(\text{train})}$ depends on the problem at hand.

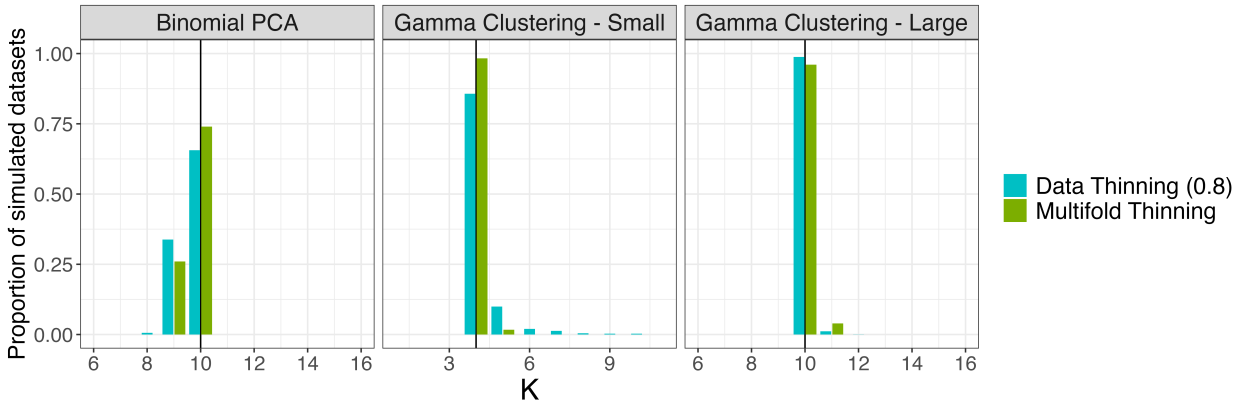

Figure 10: The proportion of simulated data sets in which each candidate value of K is selected, with the mean squared error loss, under data thinning with $\epsilon^{(\text{train})} = 0.8$ (blue) and multifold thinning with $M = 5$ (green), for each of the simulation settings described in Section 6.1. The true value of K^* is indicated by the vertical black line. Multifold thinning tends to select the true value of K more often than single-fold thinning.

(2) Log normalization: Next, the data are normalized and log transformed, such that

$$Y_{ij} = \log\left(\frac{X_{ij}}{\sum_{t=1}^{13714} X_{it}} \times 10,000 + 1\right).$$

(3) Feature selection: Following this transformation, the top 2000 highly variable genes are selected using the function `FindVariableFeatures` from the Seurat package. The goal of the function is to find a subset of features with high cell-to-cell variation after accounting for the inherent mean-variance relationship, as these are most likely to be interesting in downstream analysis, and it implements methodology from Stuart et al. (2019).

(4) Centering and scaling: Finally, the columns of the subsetted matrix $Y \in \mathbb{R}^{2638\times 2000}$ are centered and scaled to obtain the matrix $\tilde{Y}$.

After these preprocessing steps, the principal components of $\tilde{Y}$ are computed.

We now explain the preprocessing for $X^{(1)}$ and $X^{(2)}$ that we use for our Poisson data thinning alternative to the Seurat tutorial. We follow the same four steps as above, but we are careful to specify what we do on the training set $X^{(1)}$ as opposed to the test set $X^{(2)}$.

(1) Initial data filtering: We perform the initial data filtering from Step (1) above on $X^{(1)}$. We then subset $X^{(2)}$ to include the same genes and cells as those in $X^{(1)}$. After this step, $X^{(1)}$ and $X^{(2)}$ are both in $\mathbb{Z}_{\geq 0}^{2638\times 13258}$.

(2) Log normalization: We normalize and log-transform both $X^{(1)}$ and $X^{(2)}$, such that:

$$Y_{ij}^{(1)} = \log\left(\frac{X_{ij}^{(1)}}{\sum_{t=1}^{13258} X_{it}^{(1)}} \times 10,000 + 1\right), Y_{ij}^{(2)} = \log\left(\frac{X_{ij}^{(2)}}{\sum_{t=1}^{13258} X_{it}^{(2)}} \times 10,000 + 1\right).$$

We note that these random variables are still independent and identically distributed under our Poisson assumption.

(3) Feature selection: We then apply the Seurat function `FindVariableFeatures` to the matrix $Y^{(1)}$ to select the top 2000 highly variable genes (Stuart et al., 2019). We subset both $Y^{(1)}$ and $Y^{(2)}$ to contain only these genes, such that $Y^{(1)}, Y^{(2)} \in \mathbb{R}^{2638\times 2000}$.

(4) Centering and scaling: We center and scale the columns of the subsetted $Y^{(1)}$ to obtain $\tilde{Y}^{(1)}$. We also center and scale the columns of the subsetted $Y^{(2)}$ to obtain $\tilde{Y}^{(2)}$.

After these preprocessing steps, the principal components of $\tilde{Y}^{(1)}$ are computed, and the loss function is computed using $\tilde{Y}^{(2)}$.

The negative binomial data thinning analysis, presented in Figure 7(d), follows exactly the same steps as above, but the Poisson data thinning step is replaced with negative binomial data thinning. Details are given in Appendix F.3.

F.2 Equivalence Between Standard Deviation and Sum of Squared Errors

We now explain the identity that makes Figure 7(a) and Figure 7(b) mathematically equivalent. In Section 7, we defined

$$SSE_K(\tilde{Y}) = \left\|\tilde{Y} - U_{1:K}D_{1:K}V_{1:K}^T\right\|_F^2.$$

We see that:

$$\left\|\tilde{Y} - U_{1:K}D_{1:K}V_{1:K}^T\right\|_F^2 = \|\tilde{Y}\|_2^2 - 2\text{trace}\left(\tilde{Y}^T U_{1:K}D_{1:K}V_{1:K}^T\right) + \text{trace}\left(V_{1:K}D_{1:K}^T U_{1:K}^T U_{1:K}D_{1:K}V_{1:K}^T\right)$$
$$= \|\tilde{Y}\|_F^2 - \text{trace}\left(D_{1:K}^T D_{1:K}\right) = \left\|\tilde{Y}\right\|_F^2 - \sum_{j=1}^K D_{jj}^2.$$

Thus, if we compute $SSE_K(\tilde{Y})$ for $K = 1, \ldots, 20$, since $\left\|\tilde{Y}\right\|_F^2$ is fixed, we can easily obtain the values of $\sum_{j=1}^K D_{jj}^2$ for $K = 1, \ldots, 20$. By taking the differences between these values for K and $K+1$, we obtain D_{KK} for $K = 1, \ldots, 20$. Finally, we note that the standard deviation of the Kth principal component ($U_K D_{KK}$) can be written as

$$\sqrt{(D_{KK}U_K)^T(D_{KK}U_K)} = \sqrt{D_{KK}^2}.$$

Since the standard deviation of the Kth principal component (plotted in Figure 7(a)) can be obtained directly from the sums of squared error plotted in Figure 7(b), we say that the two plots are mathematically equivalent.

F.3 Negative Binomial Thinning

For the negative binomial analysis in Section 7, we assume that $X_{ij} \sim \text{NegBin}(\mu_{ij}, b_j)$, where here we adopt the parameterization such that $E[X_{ij}] = \mu_{ij}$ and $\text{Var}(X_{ij}) = \mu_{ij} + \frac{\mu_{ij}^2}{b_j}$. We refer to b_j as the overdispersion parameter, and we note that it is equivalent to the size parameter r in the parameterization of the negative binomial distribution used in the rest of this paper (e.g. Table 2 and Proposition 11). As in Choudhary and Satija (2022), we assume that the overdispersion parameters b_j are gene-specific, but not cell specific.

In order to use negative binomial thinning (Table 2), rather than Poisson thinning, to obtain $X^{(1)}$ and $X^{(2)}$, we need to estimate the gene-specific overdispersion parameters b_j. We do this following the methodology of Hafemeister and Satija (2019), implemented in the `R` package `sctransform`, which assumes a smooth relationship between the average expression for each gene and the overdispersion parameters for each gene. As these estimates are computed using the data X, the results in Proposition 11 do not apply exactly. However, the results in Proposition 11 heuristically tell us that negative binomial thinning should perform reasonably well if these estimated overdispersion parameters are close to the "true" overdispersion parameters.

We refer interested readers to Neufeld et al. (2023) for more information on negative binomial data thinning in this context, as well as simulations showing that negative binomial data thinning performs well with estimated overdispersion parameters. We leave the matter of theoretical guarantees on the performance of data thinning with estimated nuisance parameters to future work.

References

Andreas Buja, Lawrence Brown, Arun Kumar Kuchibhotla, Richard Berk, Edward George, and Linda Zhao. Models as approximations ii: A model-free theory of parametric regression. *Statistical Science*, 34(4):545–565, 2019.

Fan Chen, Sebastien Roch, Karl Rohe, and Shuqi Yu. Estimating graph dimension with cross-validated eigenvalues. *arXiv preprint arXiv:2108.03336*, 2021.

Saket Choudhary and Rahul Satija. Comparison and evaluation of statistical error models for scRNA-seq. *Genome Biology*, 23(1):1–20, 2022.

Rick Durrett. *Probability: theory and examples*, volume 49. Cambridge University Press, 2019.

Wei Fu and Patrick O Perry. Estimating the number of clusters using cross-validation. *Journal of Computational and Graphical Statistics*, 29(1):162–173, 2020.

Lucy L Gao, Jacob Bien, and Daniela Witten. Selective inference for hierarchical clustering. *Journal of the American Statistical Association*, pages 1–11, 2022.

David Gerard. Data-based RNA-Seq simulations by binomial thinning. *BMC Bioinformatics*, 21 (1):1–14, 2020.

Christoph Hafemeister and Rahul Satija. Normalization and variance stabilization of single-cell RNA-seq data using regularized negative binomial regression. *Genome biology*, 20(1):296, 2019.

Yuhan Hao, Stephanie Hao, Erica Andersen-Nissen, William M Mauck, Shiwei Zheng, Andrew Butler, Maddie J Lee, Aaron J Wilk, Charlotte Darby, Michael Zager, et al. Integrated analysis of multimodal single-cell data. *Cell*, 184(13):3573–3587, 2021.

Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, volume 2. Springer, 2009.

P. Hoffman et al. Seurat guided clustering tutorial. *https://satijalab.org/seurat/ articles/pbmc3k_tutorial.html*, 2022. Accessed: 09-12-2022.

Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning*, volume 112. Springer, 2013.

Harry Joe. Time series models with univariate margins in the convolution-closed infinitely divisible class. *Journal of Applied Probability*, 33(3):664–677, 1996.

Bent Jørgensen. Exponential dispersion models and extensions: A review. *International Statistical Review/Revue Internationale de Statistique*, pages 5–20, 1992.

Bent Jørgensen and Peter Xue-Kun Song. Stationary time series models with exponential dispersion model margins. *Journal of Applied Probability*, 35(1):78–92, 1998.

James Leiner, Boyan Duan, Larry Wasserman, and Aaditya Ramdas. Data fission: splitting a single data point. *Journal of the American Statistical Association*, (just-accepted):1–22, 2023.

Francisco Louzada, Pedro L. Ramos, and Eduardo Ramos. A note on bias of closed-form estimators for the gamma distribution derived from likelihood equations. *The American Statistician*, 73(2): 195–199, 2019.

Anna Neufeld, Lucy L Gao, Joshua Popp, Alexis Battle, and Daniela Witten. Inference after latent variable estimation for single-cell RNA sequencing data. *Biostatistics*, page kxac047, 2022. ISSN 1465-4644.

Anna Neufeld, Joshua Popp, Lucy L Gao, Alexis Battle, and Daniela Witten. Negative binomial count splitting for single-cell RNA sequencing data. *arXiv preprint arXiv:2307.12985*, 2023.

Natalia L Oliveira, Jing Lei, and Ryan J Tibshirani. Unbiased risk estimation in the normal means problem via coupled bootstrap techniques. *arXiv preprint arXiv:2111.09447*, 2021.

Natalia L Oliveira, Jing Lei, and Ryan J Tibshirani. Coupled bootstrap test error estimation for Poisson variables. *arXiv preprint arXiv:2212.01943*, 2022.

Art B Owen and Patrick O Perry. Bi-cross-validation of the SVD and the nonnegative matrix factorization. *The Annals of Applied Statistics*, 3(2):564–594, 2009.

Daniel G Rasines and G Alastair Young. Splitting strategies for post-selection inference. *Biometrika*, 2022. ISSN 1464-3510.

Alessandro Rinaldo, Larry Wasserman, and Max G'Sell. Bootstrapping and sample splitting for high-dimensional, assumption-lean inference. *The Annals of Statistics*, 47(6):3438 – 3469, 2019.

Abhishek Sarkar and Matthew Stephens. Separating measurement and expression models clarifies confusion in single-cell RNA sequencing analysis. *Nature Genetics*, 53(6):770–777, 2021.

Rahul Satija, Jeffrey A Farrell, David Gennert, Alexander F Schier, and Aviv Regev. Spatial reconstruction of single-cell gene expression data. *Nature Biotechnology*, 33:495–502, 2015.

Tim Stuart, Andrew Butler, Paul Hoffman, Christoph Hafemeister, Efthymia Papalexi, William M Mauck III, et al. Comprehensive integration of single-cell data. *Cell*, 177:1888–1902, 2019.

Xiaoying Tian. Prediction error after model search. *The Annals of Statistics*, 48(2):763–784, 2020.

Xiaoying Tian and Jonathan Taylor. Selective inference with a randomized response. *The Annals of Statistics*, 46(2):679–710, 2018.

Jingshu Wang, Mo Huang, Eduardo Torre, Hannah Dueck, Sydney Shaffer, John Murray, et al. Gene expression distribution deconvolution in single-cell RNA sequencing. *Proceedings of the National Academy of Sciences*, 115(28):E6437–E6446, 2018.

Zhi-Sheng Ye and Nan Chen. Closed-form estimators for the gamma distribution derived from likelihood equations. *The American Statistician*, 71(2):177–181, 2017.