A Bayesian Perspective on Shuffled Linear Regression

Samy Braik Éloi Munoz Malo Ruellan

2024-2025

Contents

Introduction

Linear regression with shuffled labels is the problem of performing a linear regression fit on datasets whose labels are unknowingly shuffled with respect to their inputs. The task of both recovering the parameters and identifying the permutations is inherently difficult. This is largely due to the number of possible permutations, leading to combinatorial complexity, and the loss of information and corruption caused by mismatched labels. In the following course, we will introduce several approaches to this problem with a focus on recovering both the model parameters and the associated permutation. We begin by introducing the model, as well as several real-life examples and use-cases, and motivating the need for certain assumptions, particularly the sparsity condition on the permutation. Next, we outline the main frequentist results before transitioning to the Bayesian perspective. We introduce basic Bayesian tools as well as far more uncommon one such as coarsened posteriors. Finally, we end up the course with several simulations highlighting the discussed approach.

This course is mostly based on [4].

1. Model

We assume that we have access to data $(X_1, Y_1), \dots, (X_n, Y_n)$ where $(X_i, Y_i) \in \mathbb{R}^d \times \mathbb{R}$. Some X_i values may be incorrectly paired with non-corresponding Y_i values. We will later make the assumption that the number of such mismatches is at most $k \ll n$, and more specifically, we will motivate the importance of this assumption.
There exists an unknown permutation φ on $\{1, \dots, n\}$ such that the pairs $(X_i, Y_{\varphi(i)})$ follow the standard linear regression model $Y_i = \beta^\mathsf{T} X_i + \varepsilon_i$, with $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$ independent Gaussian noise.
Let Π denote the matrix representation of the permutation φ. The model can then be expressed as

$$Y = \Pi X\beta + \varepsilon$$

where we have defined $Y = (Y_1, \dots, Y_n)^\mathsf{T}$, $X = (X_1, \dots, X_n)^\mathsf{T}$ and $\varepsilon = (\varepsilon_1, \dots, \varepsilon_n)^\mathsf{T}$. We wish to accurately infer the parameters β and σ^2 of the model from the mismatched pairs (X_i, Y_i).

We can quantify the number of mismatches using the Hamming distance between the permutation matrix Π and the identity matrix $\mathbf{I}_n$ defined as

$$d_\mathsf{H}(\Pi, \mathbf{I}_n) = \#\left\{1 \leqslant i \leqslant n \ ; \ \Pi_{i,i} = 0\right\}$$

In the following, we will say that a permutation matrix Π is k-sparse if $d_\mathsf{H}(\Pi, \mathbf{I}_n) \leqslant k$. Furthermore, let us denote the set of permutation matrices in $\mathcal{M}_n(\mathbb{R})$ by $\mathcal{P}_n$, and define the set of k-sparse matrices $\mathcal{P}_{n,k} \subset \mathcal{P}_n$ by

$$\mathcal{P}_{n,k} = \left\{\Pi \in \mathcal{P}_n \ ; \ d_\mathsf{H}(\Pi, \mathbf{I}_n) \leqslant k\right\}$$

The first frequentist idea would be to follow an approach similar to OLS Linear Regression:

$$\underset{\beta,\Pi}{\mathsf{argmin}} \ \|Y - \Pi X\beta\|^2 \tag{1}$$

This idea will be further explored in Sections 3 and 4.

A second approach would be to compute the Maximum Likelihood estimator, that is, to solve the following optimization problem:

$$(\hat{\Pi}, \hat{\beta}, \hat{\sigma}^2) = \underset{\Pi,\beta,\sigma^2}{\mathsf{argmax}} \left[-\frac{1}{2\sigma^2}\|Y - \Pi X\beta\|^2 - \frac{n}{2}\log(\sigma^2) \right]$$

We will not cover this approach in the following course as the discreteness of the permutation matrix parameter renders it much too hard even for small k values.

A third idea would be to put aside the frequentist approach and explore the Bayesian perspective. This is the main output of [4] and it will be covered in Section 6. There are several reasons to adopt a Bayesian perspective, one of which being regularisation via prior choice.

2. Examples

One of the most common scenarios in which permuted data arises is record linkage. This process involves merging records from two (or more) data sources. Practical examples could be linking the electronic health records of patients from health practitioners. If we lack unique identifiers for each data sources, we may end up with mismatches. Now, retrieving each patient's full health record is the goal but in this context the privacy problems arises too. Why would we need to combine datasets if the process is that prone to errors? The main reason is that a single dataset rarely contains all variables of interest.

Flow cytometry is another great example (as presented in [1]). Particles are suspended in a fluid and we are interested in measuring the physical and chemical characteristics of those particles, which we could refer here as labels. To achieve that, we ideally need to flow particles one at a time through a laser beam in order to best infer the particles' characteristics by looking at the scattering of the light. However, the order of the cells passing through the beam is unknown. Therefore, we end up with a dataset where labels are shuffled, motivating the need for new inference techniques.

Another interesting example, albeit somewhat unethical, would be [6].

3. Sparsity

The following figures are a starting point to understand why the sparsity assumption is crucial. In Figure 1, we see that the least-squares estimator fails to capture the true relationship between X and Y.

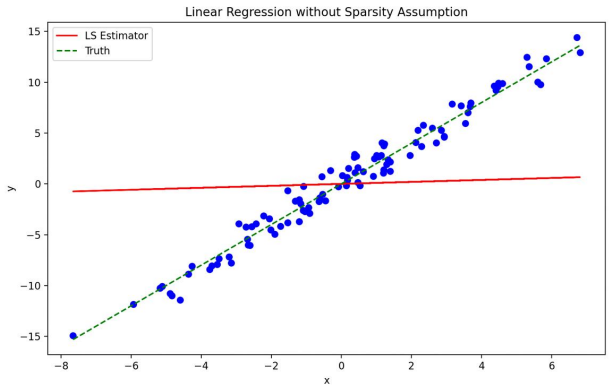

Figure 1: Least-squares estimator vs. truth.

In Figure 2, we see that permutating the labels can be highly perturbative. Without sufficient correctly identified labels, it is practically hopeless to recover the ground truth.

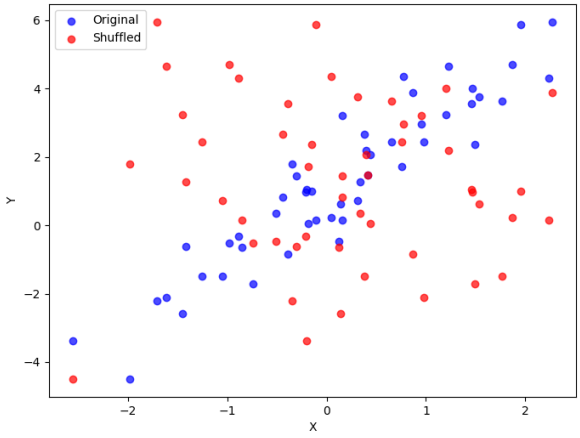

Figure 2: Original data points vs. shuffled data points.

In order to motivate the sparsity assumption related to the number of permuted labels, we present two key results that support this hypothesis.
The first result, from [1], proves the inconsistency of the least-squares estimator for a toy model. In addition, we provide the limit of the estimator in this specific case.

Proposition 1.
Let $X \sim \mathcal{N}(0, \kappa^2 \mathbf{I}_n)$ and $Y = \Pi X\beta + \varepsilon$ with an unknown scalar weight $\beta > 0$, an unknown $n \times n$ permutation matrix Π and additive noise $\varepsilon \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_n)$. Then, the least-squares estimate of β is inconsistent. More precisely, the estimator converges almost surely to the following limit:

$$\lim_{n \to +\infty} \hat{\beta}_{\mathsf{LS}} = \beta \cdot \sqrt{1 + \frac{\sigma^2}{\kappa^2 \beta^2}}$$

The proof of Proposition 1 requires the use of the following lemmas. Note that the proof of Lemma 1 uses a similar argument to that of the rearrangement inequality.

Lemma 1.
For $d = 1$ and $\beta > 0$, the least-squares estimator can be written as follows:

$$\hat{\beta}_{\mathsf{LS}} = \operatorname*{argmin}_{\beta} |Y_{\uparrow} - \beta X_{\uparrow}|^2$$

where, for $v \in \mathbb{R}^n$, the notation $v_{\uparrow}$ denotes the vector that consists of the n coordinates of v sorted in ascending order.

Remark. We have assumed for simplicity that $\beta > 0$. If $\beta < 0$, a similar argument can be made to show that the least-squares difference occurs when X is sorted in descending order.

Proof of Lemma 1.
Recall that the least-squares estimator is defined as:

$$\hat{\beta}_{\mathsf{LS}} = \operatorname*{argmin}_{\beta,\Pi} \|Y - \beta \Pi X\|^2$$

We claim that, for $d = 1$ and $\beta > 0$, the following optimization problems are equivalent:

$$\operatorname*{argmin}_{\beta} \min_{\Pi} |Y - \beta \Pi X|^2 = \operatorname*{argmin}_{\beta} |Y_{\uparrow} - \beta X_{\uparrow}|^2$$

Write $Z = \beta X$. Since $\beta > 0$, we have $Z_{\uparrow} = \beta X_{\uparrow}$. We will show that $|Y_{\uparrow} - \Pi Z|^2$ is minimal when Z is sorted in ascending order.
Assume that $|Y_{\uparrow} - \Pi Z|^2$ is minimized for a permutation Z' of Z that does not have its entries in ascending order. Then, there must be i, j such that $i < j$ and $Z'_i > Z'_j$.
We single out the following terms:

$$(Y_i^{\uparrow} - Z'_i)^2 + (Y_j^{\uparrow} - Z'_j)^2 \geqslant (Y_i^{\uparrow} - Z'_j)^2 + (Y_j^{\uparrow} - Z'_i)^2$$

Since $Y_{\uparrow}$ is sorted in ascending order and $Z'_i > Z'_j$ by assumption, the difference is:

$$2(Y_i^{\uparrow} Z'_i + Y_j^{\uparrow} Z'_j - Y_i^{\uparrow} Z'_j - Y_j^{\uparrow} Z'_i) = 2(Y_j^{\uparrow} - Y_i^{\uparrow})(Z'_i - Z'_j) \geqslant 0$$

This contradicts the minimality of $|Y_{\uparrow} - Z'|^2$.

Lemma 2.
Let $X \sim \mathcal{N}(0, \kappa^2 \mathbf{I}_n)$ and $Y \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_n)$. Then, we have the following limit:

$$\frac{1}{n} \sum_{i=1}^{n} X_{(i)} Y_{(i)} \xrightarrow{\mathsf{P}} \kappa \cdot \sigma$$

as $n \to +\infty$.

Proof of Lemma 2.

We will make use of Lemma 3 which can be found below.

Consider independent and identically distributed Gaussian random variables $Z_1, \dots, Z_n$ and $Z'_1, \dots, Z'_n$.

For all $\delta > 0$ and $i \geqslant 1$, combining Lemma 3 and Chebyshev's inequality, we get:

$$\mathbb{P}\big(|Z_{(i)} - \mathbb{E}(Z_{(i)})| > \delta\big) \leqslant \frac{\mathsf{Var}(Z_{(i)})}{\delta^2} \to 0$$

The triangle inequality gives the following decomposition:

$$\begin{aligned}\left\{|Z_{(i)} - Z'_{(i)}| > \delta\right\} &\subset \left\{|Z_{(i)} - \mathbb{E}(Z_{(i)})| + |Z'_{(i)} - \mathbb{E}(Z'_{(i)})| > \delta\right\} \\ &\subset \left\{|Z_{(i)} - \mathbb{E}(Z_{(i)})| > \delta/2\right\} \cup \left\{|Z'_{(i)} - \mathbb{E}(Z'_{(i)})| > \delta/2\right\}\end{aligned}$$

which leads to:

$$\mathbb{P}\big(|Z_{(i)} - Z'_{(i)}| > \delta\big) \leqslant \mathbb{P}\big(|Z_{(i)} - \mathbb{E}(Z_{(i)})| > \delta/2\big) + \mathbb{P}\big(|Z'_{(i)} - \mathbb{E}(Z'_{(i)})| > \delta/2\big) \to 0 \tag{2}$$

We can write X_i and Y_i as $X_i = \kappa Z_i$ and $Y_i = \sigma Z'_i$.

Since multiplying by a positive constant does not affect the ordering of elements in a vector, we can rewrite the expression as:

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^{n} X_{(i)}Y_{(i)} = \frac{1}{n}\sum_{i=1}^{n} \kappa Z_{(i)} \cdot \sigma Z'_{(i)} &= \frac{\kappa\sigma}{n}\sum_{i=1}^{n} Z_{(i)}Z'_{(i)} \\ &= \frac{\kappa\sigma}{2n}\sum_{i=1}^{n}\left[Z^2_{(i)} + Z'^2_{(i)} - (Z_{(i)} - Z'_{(i)})^2\right] \end{aligned} \tag{3}$$

The law of large numbers gives:

$$\frac{1}{n}\sum_{i=1}^{n} Z^2_{(i)} = \frac{1}{n}\sum_{i=1}^{n} Z^2_i \to 1 \tag{4}$$

as $n \to +\infty$. The same goes for the Z'_i.

Combining equations (2), (3) and (4), we get:

$$\frac{1}{n}\sum_{i=1}^{n} X_{(i)}Y_{(i)} \to \kappa \cdot \sigma$$

The following lemma can be derived from Proposition 4.2 in [7].

Lemma 3.

Let $Z_1, \dots, Z_n \sim \mathcal{N}(0,1)$. Consider the order statistics $Z_{(1)} \leqslant \cdots \leqslant Z_{(n)}$. Then, for each k, $\mathsf{Var}(Z_{(k)}) \to 0$ as $n \to +\infty$.

We can now go back to proving Proposition 1.

Proof of Proposition 1.
Using Lemma 1, we can simply apply ordinary least-squares regression to $X_\uparrow$ and $Y_\uparrow$ in order to compute $\hat{\beta}_{\mathsf{LS}}$. It is well-known that $\hat{\beta}_{\mathsf{LS}}$ has the following closed-form:

$$\hat{\beta}_{\mathsf{LS}} = \left[\frac{1}{n} X_\uparrow^\mathsf{T} X_\uparrow\right]^{-1} \left[\frac{1}{n} X_\uparrow^\mathsf{T} Y_\uparrow\right] = \left[\frac{1}{n}\sum_{i=1}^{n} X_{(i)}^2\right]^{-1} \left[\frac{1}{n}\sum_{i=1}^{n} X_{(i)} Y_{(i)}\right]$$

Using Slutsky's theorem, we can find the limits of these terms separately. The first term simply requires the law of large numbers:

$$\frac{1}{n}\sum_{i=1}^{n} X_{(i)}^2 = \frac{1}{n}\sum_{i=1}^{n} X_i^2 \to \mathbb{E}[X_1^2] = \kappa^2$$

The limit of the second term is covered by Lemma 2 with $X \sim \mathcal{N}(0, \kappa^2 \mathbf{I}_n)$ and $Y = \Pi X\beta + \varepsilon$ is a Gaussian vector with $\mathbb{E}[Y] = 0$ and $\mathsf{Var}(Y) = \beta^2 \Pi \mathsf{Var}(X)\Pi^\mathsf{T} + \mathsf{Var}(\varepsilon) = (\beta^2\kappa^2 + \sigma^2)\mathbf{I}_n$. (Permutation matrices are orthogonal.) Combining both limits gives the desired result.

The following proposition, from [3], claims that in a pure noise setting with $\Pi = \mathbf{I}_n$, the least-squares estimator can be bounded away from zero.

Proposition 2.
Let $\beta = 0$ and $\Pi = \mathbf{I}_n$, i.e. $Y = \varepsilon \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_n)$ and consider the least-squares estimator $\hat{\beta}_{\mathsf{LS}}$ as defined in (1) with $k = n$. Then, there exist constants $c_1, c_2 > 0$ such that with probability at least $1 - c_1 e^{-c_2 n}$:

$$\|\hat{\beta}_{\mathsf{LS}}\|_2^2 \geqslant \frac{\sigma^2}{32\pi^2} \cdot \frac{n}{2n + d}$$

This is a striking example of how the true model parameters may not be recovered even when no actual shuffling has occurred.

4. Main Frequentist Results

In this section, we delve into the frequentist approach to our problem, presenting a method for constructing an estimator of (Π, β) and examining its performance.
With previous section's results in mind, it is natural to assume the sparsity on the Π matrix. This assumption will be maintained throughout the remainder of this course. Leveraging this new condition, [3] proposes a new method to build a good estimator of β. By relaxing multiple times a NP-hard minimisation problem, we end up with an easier formulation:

$$\min_{\beta \in \mathbb{R}^d,\, e \in \mathbb{R}^n} \frac{1}{n}\|Y - X\beta - \sqrt{n}e\|_2^2 + \lambda \|e\|_1, \quad \lambda > 0 \tag{5}$$

We proceed by presenting the theorem that establishes a bound on the estimation error of the true parameter β^* given a solution of (5).

Theorem 1.
Let $(\widetilde{\beta}, \widetilde{e})$ be a minimizer of (5) with $\lambda = 4(1+M)\sigma\sqrt{2\log(n)/n}$ for some $M > 0$.
There exist constants c_1, c_2, M, ε so that if $k \leq c_1 \dfrac{n-d}{\log(n/k)}$ the following inequalities hold with probability at least $1 - 2\exp(-c_2(n-d)) - 2n^{-M^2} - 2\exp(-(d \vee \log(n))/2)$

$$\|\widetilde{\beta} - \beta^*\|_2 \leq \frac{\sigma}{1 - \sqrt{\dfrac{4d \vee \log(n)}{n}}} \left(\sqrt{\frac{5(d \vee \log(n))}{n}} + 48(1+M)\frac{n}{n-d}\varepsilon^{-1}\sqrt{\frac{2k\log(n)}{n}} \right)$$

The first component of the bound is an incompressible error, which would still be present even if Π^* were known. In contrast, the second component exists solely due to the excess error introduced by the unknown parameter Π^*.
Given an accurate estimate of β^*, we can leverage it to construct an estimator of Π^* that is computationally feasible. The method is straightforward, we use the same naive method we would use if β^* were known and replace it by its estimator $\widetilde{\beta}$. This leads to the following problem, which is an integer linear problem.

$$\begin{aligned} &\underset{\Pi \in \mathcal{P}_n}{\mathsf{argmin}} \ \langle \Pi X\beta, Y \rangle \\ &\text{subject to } \ d_{\mathsf{H}}(\Pi, \mathbf{I}_n) \leqslant k. \end{aligned} \tag{6}$$

5. Bayesian Inference

In this section, we will introduce the main Bayesian concepts and ideas as opposed to the frequentist approach. Bayesian inference combines prior belief with data to obtain posterior distributions on which one can perform statistical inference. Except for some simple cases (for example, conjugate distributions), Bayesian inference can be computationally intensive and may rely on computational techniques such as Markov Chain Monte Carlo. In [4], the authors outright skip part of these issues by considering coarsened posteriors as presented in Section 6. The reader may refer to [2] and [5] for more details. Firstly, we give a broad definition of a statistical experience.

Definition. *Statistical Experience*
A statistical experience is given by:

(*i*) A random variable X defined on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ with values in a measurable space $(E, \mathcal{E})$.

(*ii*) A set Θ referred to as the parameter space. In the following, we will have $\Theta \subseteq \mathbb{R}^d$.

(*iii*) A family of probability measures $\mathcal{P}$ on $(E, \mathcal{E})$ referred to as the model :

$$\mathcal{P} = \Big\{ P_\theta \, ; \, \theta \in \Theta \Big\}$$

5.1. Frequentist Approach

Frequentist inference relies on choosing a model $\mathcal{P}$ with the assumption that the distribution of X belongs to $\mathcal{P}$. In other words, we make the assumption that there exists $\theta \in \Theta$ such that $X \sim P_\theta$. We then wish to estimate θ with samples of the random variable X. We distinguish several major questions:

(*i*) **Estimation.** Given observations $X_1, \dots, X_n \overset{d}{=} X$. Constructing an estimator $\hat{\theta} = \hat{\theta}(X_1, \dots, X_n)$ of θ with nice properties.

(*ii*) **Confidence intervals.** Formulating a random subset $\mathcal{R} = \mathcal{R}(X_1, \dots, X_n)$ of Θ such that $\theta \in \mathcal{R}$ with high probability (under P_θ).

(*iii*) **Tests.** Answer with high probability whether θ verifies a certain property given some observations by constructing a test $\varphi = \varphi(X_1, \dots, X_n)$ with values in $\{0, 1\}$.

5.2. Bayesian Approach

Bayesian inference suggests we model unknown parameters with random variables. We assume that we have some prior knowledge on the underlying properties of our parameters that we infuse in our model with a distribution on the parameters, named the prior distribution. Observations allow us to update this distribution by creating the posterior distribution. Intuitively, this entails confronting our beliefs with the reality in order to formulate some posterior appreciation.

As such, we assume that the unknown parameter $\boldsymbol{\theta}$ is random (which we write in bold to distinguish with the frequentist case) with distribution Π and $\mathsf{supp}(\Pi) \subseteq \Theta$.

$$\boldsymbol{\theta} \sim \Pi, \quad \mathbf{X} = (X_1, \dots, X_n) \,|\, \boldsymbol{\theta} \sim P_{\boldsymbol{\theta}}^{\otimes n}$$

The posterior distribution is given by $\Pi(\mathbf{X} \,|\, \boldsymbol{\theta})$.

Remark. P_θ is no longer the distribution of X, but the distribution of $X \,|\, \boldsymbol{\theta} = \theta$.

From now on, we assume the following:

(*i*) For all $\theta \in \Theta$, P_θ has a density p_θ with respect to the same σ-finite measure μ on $(E, \mathcal{E})$.

(*ii*) The distribution Π has a density π with respect to a positive σ-finite measure ν on Θ.

Theorem 2. *Bayes' Formula*
The density of the posterior distribution $\Pi(\boldsymbol{\theta} \,|\, \mathbf{X})$, with regard to ν, is given by:

$$\forall \theta \in \Theta, \ \pi(\theta \,|\, \mathbf{X}) = \frac{\pi(\theta) p_\theta^{\otimes n}(\mathbf{X})}{f(\mathbf{X})}$$

Here, $f(\mathbf{X})$ is the following normalizing constant:

$$f(\mathbf{X}) = \int_\Theta \pi(\theta) p_\theta^{\otimes n}(\mathbf{X}) \, d\nu(\theta)$$

Remark. In practice, we disregard the normalizing constant and use $\pi(\theta \mid \mathbf{X}) \propto \pi(\theta) p_\theta(\mathbf{X})$. Indeed, the normalizing constant may be very complicated to derive. In this context, the Markov Chain Monte Carlo (as presented further below) holds a very prominent place in the Bayesian toolkit.

The following definition presents a special case where the posterior has a closed-form and allows for computational ease.

Definition. *Conjugate Families*
Given a sampling distribution and a prior distribution, if the resulting posterior distribution belongs to the same parametric family of distributions than the prior distribution, then we say that the prior distribution is a conjugate prior for this sampling distribution.

5.3. Markov Chain Monte Carlo

Assume that we wish to sample from a probability distribution P with density p. Further assume that we know how to sample from a family of distributions $Q(x, \cdot)$ with densities $q(x, \cdot)$. The Metropolis-Hastings algorithm is based on ergodic theorems and enables us to iteratively construct samples distributed (asymptotically) according to P. We succinctly describe how it works below.

We start with an initial value X_0. At iteration $t+1$, we readily have a simulation X_t. We sample $Y \sim Q(X_t, \cdot)$. We then accept this transition with probability

$$r(X_t, Y) = \frac{p(Y) q(Y, X_t)}{p(X_t) q(X_t, Y)} \wedge 1.$$

We can, for instance, sample $U_{t+1} \sim \mathsf{Unif}(0,1)$ and define X_{t+1} as

$$X_{t+1} = Y \mathbf{1}(U_{t+1} \leqslant r(X_t, Y)) + X_t \mathbf{1}(U_{t+1} > r(X_t, Y)).$$

Remark. As mentioned earlier, the Metropolis-Hastings reveals to be very interesting for Bayesian practicioners as it is only required to know the densities up to a multiplicative constant.

Remark. The Metropolis-Hastings algorithm offers no theoretical guarantee of convergence in finite time. One may use the Gelman-Rubin or Geweke diagnostics to test for convergence.

6. Bayesian Results

6.1. Model and Prior Specification

Thereafter, we come back to the setup outlined in Section 1.
Suppose we observe data $\mathcal{D}_n = \left\{ (X_1, Y_1), \dots, (X_n, Y_n) \right\}$ with $(X_i, Y_i) \in \mathbb{R}^d \times \mathbb{R}$, and at most $k \ll n$ mismatches.
We consider the following model

$$Y = \Pi X \beta + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_n)$$

where the permutation matrix $\Pi = (\pi_{i,j})$ is assumed to be such that

$$d_{\mathsf{H}}(\Pi, \mathbf{I}_n) = \sum_{i=1}^{n} |\pi_{i,i} - 1| \leqslant k.$$

The objective is to develop a Bayesian framework to infer parameters Π and (β, σ^2). Therefore, we start by specifying priors on the model parameters. The authors choose $\Pi \sim \mathsf{Unif}(\mathcal{P}_{n,k})$. In other words, we assume

$$p(\Pi) = \frac{1}{|\mathcal{P}_{n,k}|}$$

for $\Pi \in \mathcal{P}_{n,k}$. We choose a non-informative prior on Π for simplicity, but one may specify more elaborate priors while still imposing the restriction $\Pi \in \mathcal{P}_{n,k}$. Furthermore, the authors choose the following priors for β and σ^2

$$\beta \sim \mathcal{N}(0, 1000\mathbf{I}_d), \quad \sigma^2 \sim \mathcal{N}^+(0, 1000)$$

where $\mathcal{N}^+(\mu, \kappa^2)$ denotes the normal distribution with mean μ and variance κ^2 truncated to positive values only. Other priors for β and σ^2 could be investigated, as long as $\sigma^2 > 0$. For instance, one could use (as is fairly common) a Gamma distribution as a prior for σ^2.

Remark. Here, the authors assign large variances in order to keep priors fairly uninformative and adapted to various datasets.

Proposition 3.
Under the model and prior specification defined above, the joint posterior of (β, σ^2, Π) can be expressed as

$$p(\beta, \sigma^2, \Pi \,|\, \mathcal{D}_n) \propto \frac{1}{(\sigma^2)^{n/2}} \exp\left(-\frac{\|Y - \Pi X\beta\|^2}{2\sigma^2}\right) p(\beta)p(\sigma^2)p(\Pi)$$

where priors, posteriors, joint and marginal densities are all denoted by p for simplicity and (hopefully) without risk of confusion.

Proof of Proposition 3.
Using Bayes' formula, we get:

$$\begin{aligned} p(\beta, \sigma^2, \Pi \,|\, \mathcal{D}_n) &\propto p(Y \,|\, \beta, \sigma^2, \Pi)p(\beta)p(\sigma^2)p(\Pi)p(Y \,|\, \beta, \sigma^2, \Pi) \\ &\propto \frac{1}{(\sigma^2)^{n/2}} \exp\left(-\frac{\|Y - \Pi X\beta\|^2}{2\sigma^2}\right) p(\beta)p(\sigma^2)p(\Pi) \end{aligned}$$

since $Y \,|\, (\beta, \sigma^2, \Pi) \sim \mathcal{N}(\Pi X\beta, \sigma^2\mathbf{I}_n)$.

6.2. Robust Bayesian Approach

Sampling from the conditional distribution $\Pi \,|\, (\beta, \sigma^2)$ poses a difficult combinatorial problem. Chakraborty et al.[4] propose instead to develop a robust Bayesian approach by treating the permuted data as outliers.

In [5], Miller & Dunson advocate for the use of coarsened posteriors to enhance the robustness of Bayesian inference against model perturbations. They suggest conditioning on a neighbourhood of the empirical distribution rather than directly on the data. Moreover, they demonstrate that this approach results in a coarsened posterior that can be approximated by a tempered likelihood: the likelihood raised to a fractional power.

Generally, in a Bayesian setting, one conditions on the event that the observed data is generated from the true data generating process to define the posterior. Here, we will condition on the event that the observed data is generated by a mechanism close to the true data generating process. This will allow us to define a more robust posterior by allowing for small perturbations in the data generating mechanism. We clarify below what we mean by perturbation.

Let $\mathcal{D}_n^\star = (X_i^\star, Y_i^\star)_{i=1}^n$ denote the unobserved uncorrupted dataset, identically and independently generated from the linear regression model. The observed sparsely permuted dataset $\mathcal{D}_n$ is actually a mildly corrupted version of $\mathcal{D}_n^\star$ in the sense that $\mathsf{D}(\hat{\mathsf{P}}_\star, \hat{\mathsf{P}}) < r$ for some statistical distance D and some $r > 0$, where

$$\hat{\mathsf{P}} = \frac{1}{n}\sum_{i=1}^n \delta_{(X_i, Y_i)} \text{ and } \hat{\mathsf{P}}_\star = \frac{1}{n}\sum_{i=1}^n \delta_{(X_i^\star, Y_i^\star)}$$

respectively denote the empirical distributions with respect to $\mathcal{D}_n$ and $\mathcal{D}_n^\star$.

Without any permutation, we could use the standard posterior. However, due to the disruption stemming from the permutation, we condition on the event that $\mathsf{D}(\hat{\mathsf{P}}_\star, \hat{\mathsf{P}}) < r$. In other words, we consider the coarsened posterior $\tilde{p}(\beta, \sigma^2, \Pi \,|\, \mathsf{D}(\hat{\mathsf{P}}_\star, \hat{\mathsf{P}}) < r)$. Furthermore, we specify an exponential prior on r with mean $1/\kappa$, independently of (β, σ^2, Π) and $\mathcal{D}_n$.
It is then possible to show that we have the following approximation

$$\tilde{p}(\beta, \sigma^2, \Pi \,|\, \mathsf{D}(\hat{\mathsf{P}}_\star, \hat{\mathsf{P}}) < r) \approx \exp\Big(-\kappa \mathsf{D}(\hat{\mathsf{P}}_\star, \hat{\mathsf{P}}) < r)\Big) p(\beta) p(\sigma^2) p(\Pi)$$

up to a multiplicative constant.
Moreover, we specify the statistical discrepancy D as the Kullback-Leibler divergence (or relative entropy) defined by

$$\mathsf{KL}(p, q) = \int p(x) \log\left(\frac{p(x)}{q(x)}\right) dx$$

Then, the approximation of the coarsened posterior reduces to

$$\tilde{p}(\beta, \sigma^2, \Pi \,|\, \mathsf{KL}(\hat{\mathsf{P}}_\star, \hat{\mathsf{P}}) < r) \approx \frac{1}{(\sigma^2)^{n\alpha/2}} \exp\left(-\frac{\alpha \|Y - \Pi X \beta\|^2}{2\sigma^2}\right) p(\beta) p(\sigma^2) p(\Pi)$$

up to a multiplicative constant and for some $\alpha \in (0, 1)$ that depends on κ. In the following, we treat α as a hyper-parameter. The authors suggest using $\alpha = 1/n$.

6.3. Posterior Sampling

In the following section, we will present an algorithm to sample from the posterior. We will omit the $\mathcal{D}_n$ throughout the section for simplification purposes. One may simply devise a Gibbs sampling

scheme to carry out the inference of parameters, by alternately sampling from $\Pi \,|\, (\beta, \sigma^2)$ and $(\beta, \sigma^2) \,|\, \Pi$. We present below the algorithm proposed by the authors in [4].

Step 1. To sample from $(\beta, \sigma^2) \,|\, \Pi$, one may use a standard Metropolis-Hastings algorithm. However, finding a transition kernel tailored to the priors on β and σ^2 may be difficult. A more novel approach would be to use the Hamiltonian Monte Carlo algorithm which is adaptive to the distribution from which we wish to sample. We refer the reader to [8] for a primer on Hamiltonian Monte Carlo.

Step 2. Sampling from $\Pi \,|\, (\beta, \sigma^2)$ poses a difficult combinatorial problem as is usually the case for complex discrete distributions. The authors propose to update the chain with the posterior mode of $\Pi \,|\, (\beta, \sigma^2)$ instead. First, given β and σ^2, we compute the $n \times n$ cost matrix C such that

$$\mathsf{C}_{i,j} = \alpha \left[\frac{(Y_i - \beta^\mathsf{T}(\Pi X)_j)^2}{2\sigma^2} + \frac{n}{2} \log(\sigma^2) \right]$$

for all i, j. Then, we solve the following constrained binary optimal transport problem:

$$\mathsf{B}_* = \underset{\mathsf{B}}{\mathsf{argmin}} \sum_{i,j} b_{i,j} \mathsf{C}_{i,j}$$

subject to $\mathsf{B} \in \mathbf{U}(\mathbf{1}_n, \mathbf{1}_n)$.
Here, $\mathbf{U}(\mathbf{1}_n, \mathbf{1}_n) = \mathcal{P}_n$ is the polytope of $n \times n$ binary matrices, that is the set of $n \times n$ binary matrices B such that $\mathsf{B}\mathbf{1}_n = \mathbf{1}_n$ and $\mathsf{B}^\mathsf{T}\mathbf{1}_n = \mathbf{1}_n$ (i.e., the set of binary matrices containing exactly once the number 1 per row and column).

7. Simulations

Our goal is to conduct full Bayesian inference of the parameters Π and (β, σ^2) via the proposed methodology, for a sample size $n = 100$, sparsity constraint $s = 6$ and the hyper-parameter $\alpha = 1/n$.

We first generate $n = 100$ observations from the standard Linear Regression model

$$Y_i = X_i^\mathsf{T} \beta + \varepsilon_i, \quad i = 1, \dots, n$$

where the $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$ are independent Gaussian noise. We set $\sigma = 0.1$, and $\beta = (1, \dots, 1)^\mathsf{T} \in \mathbb{R}^5$. We then permute the first $s = 6$ observations of $X = (X_1, \dots, X_n)$ such that:

$$1 \to 6, \ 6 \to 5, \ \dots, \ 2 \to 1$$

while keeping the other observations untouched.

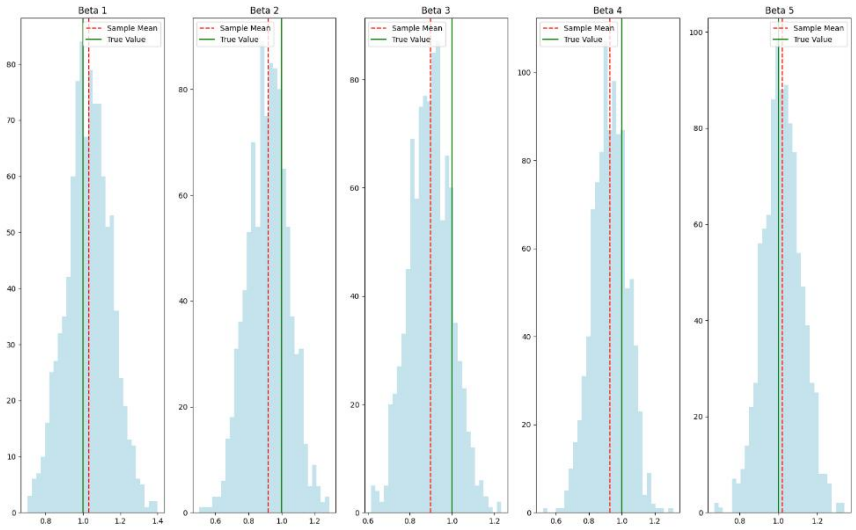

Figure 3: Posterior distributions of β for $\alpha = 1/n$.

In Figure 3, the posterior distributions of β with a sample size of $n = 100$ seem to indicate that we may accurately recover the true parameters using the sample mean.

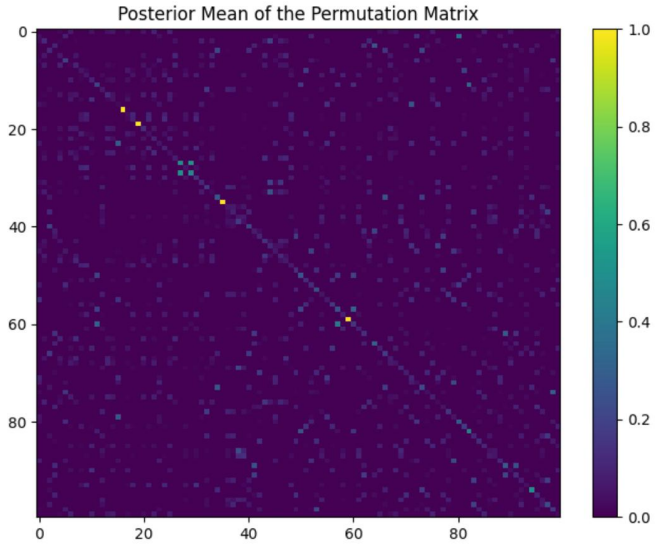

Figure 4: Posterior mean of Π for $\alpha = 1/n$.

As shown in Figure 4, the proposed methodology with $\alpha = 1/n$ achieves a good recovery of the permutation matrix Π. This highlights the effectiveness of employing coarsened posteriors in enhancing the recovery of the parameters of interest.

References

[1] J. Zou A. Abid A. Poon. *Linear Regression with Shuffled Labels*. 2017. arXiv: 1705.01342 [stat.ML]. URL: https://arxiv.org/abs/1705.01342.

[2] Y. Yang A. Bhattacharya D. Pati. *Bayesian Fractional Posteriors*. 2016. arXiv: 1611.01125 [math.ST]. URL: https://arxiv.org/abs/1611.01125.

[3] M. Slawski & E. Ben-David. *Linear Regression with Sparsely Permuted Data*. 2017. arXiv: 1710.06030 [math.ST]. URL: https://arxiv.org/abs/1710.06030.

[4] A. Chakraborty & S. Datta. *Learning with Sparsely Permuted Data: A Robust Bayesian Approach*. 2024. arXiv: 2409.10678 [math.ST]. URL: https://arxiv.org/abs/2409.10678.

[5] J. W. Miller & D. B. Dunson. *Robust Bayesian Inference via Coarsening*. 2015. arXiv: 1506.06101 [stat.ME]. URL: https://arxiv.org/abs/1506.06101.

[6] A. Narayanan & V. Shmatikov. *Robust De-Anonymization of Large Sparse Datasets*. 2008. DOI: 10.1109/SP.2008.33.

[7] S. Boucheron & M. Thomas. *Concentration Inequalities for Order Statistics*. 2012. DOI: 10.1214/ecp.v17-2210. URL: http://dx.doi.org/10.1214/ECP.v17-2210.

[8] N. K. Vishnoi. *An Introduction to Hamiltonian Monte Carlo Method for Sampling*. 2021. arXiv: 2108.12107 [cs.DS]. URL: https://arxiv.org/abs/2108.12107.

A. Code

For easy reproducibility of our simulated results, we provide the following source code written in Python.

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import linear_sum_assignment
from scipy.stats import invgamma, multivariate_normal

def generate_data(n, beta, sigma):
    d = len(beta)
    X = np.random.randn(n, d)
    noise = np.random.normal(0, sigma, n)
    y = X @ beta + noise
    return X, y

def apply_permutation(X, s0):
    permuted_X = X.copy()
    PId = np.eye(n)
    if s0 > min(X.shape):
        raise ValueError("s0 cannot be larger than the dimensions of the matrix.")
    PId[:s0, :s0] = np.flip(np.eye(s0), axis=1)
    permuted_X = np.dot(PId, X)
    return permuted_X

def compute_cost_matrix(y, X, beta, sigma_squared):
    n = len(y)
    cost_matrix = np.zeros((n, n))
    for i in range(n):
        for j in range(n):
            y_pred = np.dot(X[j], beta)
            cost_matrix[i, j] = (1 / n) * ((y[i] - y_pred) ** 2) /
            (2 * sigma_squared) + np.log(sigma_squared)
    return cost_matrix

def solve_optimal_transport(cost_matrix):
    row_ind, col_ind = linear_sum_assignment(cost_matrix)
    permutation_matrix = np.zeros_like(cost_matrix)
    permutation_matrix[row_ind, col_ind] = 1
    return permutation_matrix

def sample_beta_sigma2(y, X, permutation_matrix):
```

```
    y_permuted = np.dot(permutation_matrix, y)
    XT_X = np.dot(X.T, X)
    XT_y = np.dot(X.T, y_permuted)
    beta_mean = np.linalg.solve(XT_X, XT_y)
    beta_cov = np.linalg.inv(XT_X)
    beta_sample = multivariate_normal.rvs(mean=beta_mean, cov=beta_cov)
    residuals = y_permuted - np.dot(X, beta_sample)
    alpha = len(y) / 2
    beta_param = np.sum(residuals ** 2) / 2
    sigma2_sample = invgamma.rvs(a=alpha, scale=beta_param)
    return beta_sample, sigma2_sample

def gibbs_sampling(y, X, iterations=1000):
    n = len(y)
    d = X.shape[1]
    beta = np.zeros(d)
    sigma2 = 1.0
    permutation_matrix = np.eye(n)
    beta_samples = []
    sigma2_samples = []
    permutation_samples = []
    for _ in range(iterations):
        beta, sigma2 = sample_beta_sigma2(y, X, permutation_matrix)
        cost_matrix = compute_cost_matrix(y, X, beta, sigma2)
        permutation_matrix = solve_optimal_transport(cost_matrix)
        beta_samples.append(beta)
        sigma2_samples.append(sigma2)
        permutation_samples.append(permutation_matrix)
    return beta_samples, sigma2_samples, permutation_samples

def visualize_permutation(X, permuted_X, s0):
    fig, axes = plt.subplots(1, 2, figsize=(14, 6))
    axes[0].imshow(X[:s0], aspect='auto', cmap='viridis')
    axes[0].set_title('Original Design Matrix (First s0 Rows)')
    axes[1].imshow(permuted_X[:s0], aspect='auto', cmap='viridis')
    axes[1].set_title('Permuted Design Matrix (First s0 Rows)')
    plt.tight_layout()
    plt.show()

def compute_posterior_mean(permutations):
    n = min(600, len(permutations))
    return np.mean(permutations[-n:], axis=0)

```

```
n = 100
s0 = 7
beta = np.ones(20)
sigma = 0.1
X, y = generate_data(n, beta, sigma)
permuted_X = apply_permutation(X, s0)
visualize_permutation(X, permuted_X, s0)
beta_samples_permuted, sigma2_samples_permuted, permutation_samples_permuted =
gibbs_sampling(y, permuted_X, iterations=1000)
posterior_mean_permutation = compute_posterior_mean(permutation_samples_permuted)
plt.figure(figsize=(8, 6))
plt.imshow(posterior_mean_permutation, cmap='viridis', aspect='auto')
plt.title('Posterior Mean of the Permutation Matrix')
plt.colorbar()
plt.show()
beta_samples_array = np.array(beta_samples_permuted)
n_coordinates = beta_samples_array.shape[1]
fig, axes = plt.subplots(n_coordinates // 5, 5, figsize=(20, 12))
axes = axes.flatten()
for i in range(n_coordinates):
    beta_values = beta_samples_array[:, i]
    mean_value = np.mean(beta_values)
    axes[i].hist(beta_values, bins=30, color='lightblue', alpha=0.7)
    axes[i].axvline(mean_value, color='red', linestyle='--', label='Sample Mean')
    axes[i].axvline(beta[i], color='green', linestyle='-', label='True Value')
    axes[i].set_title(f'Beta {i + 1}')
    axes[i].legend()
plt.tight_layout()
plt.show()
```