Looking deeper into LIME

Damien Garreau DAMIEN.GARREAU@UNICE.FR
Laboratoire J. A. Dieudonné & Inria Maasai project-team
Université Côte d'Azur
Nice, France

Ulrike von Luxburg ULRIKE.LUXBURG@UNI-TUEBINGEN.DE
University of Tübingen
Department of Computer Science
Tübingen, Germany

Abstract

Interpretability of machine learning algorithm is a pressing need. Numerous methods appeared in recent years, but do they make sense in simple cases? In this paper, we present a thorough theoretical analysis of Tabular LIME. In particular, we show that the explanations provided by Tabular LIME are close to an explicit expression in the large sample limit. We leverage this knowledge when the function to explain has some nice algebraic structure (linear, multiplicative, or depending on a subset of the coordinates) and provide some interesting insights on the explanations provided in these cases. In particular, we show that Tabular LIME provides explanations that are proportional to the coefficients of the function to explain in the linear case, and provably discards coordinates unused by the function to explain in the general case.

Keywords: explainable AI, statistical learning theory

1. Introduction

Many recent progresses in machine learning came with increasingly complex models. The latest and maybe most shocking example of this trend is the recent disclosure of GPT-3 (Brown et al., 2020), a 175 *billions* parameters language model. Even though the architecture and the parameters of the model are known, in the sense that one can read the code of the model and check the value of each individual coefficient, it is very challenging to understand how a particular prediction is made.

While some users mainly care about performance (that is, accuracy), some specific applications demand *interpretability* of the algorithms involved in the decision-making process. This is in particular the case in healthcare. The main worry is that our model learns a rule yielding good accuracy on the train set, but making little common sense and leading to dramatic decisions when deployed in the wild. For instance, Caruana et al. (2015) describe a model trained to predict probability of death from pneumonia. This model ends up assigning less risk to patients if they also have asthma. Of course, from a medical point of view, this is non-sense, and deploying the model in a real-life situation would mean asthmatic patients to receive less prompt treatment and increasing their risk. One can surmise that the model learned that asthma was predictive of a lower risk because these patients, *a contrario*, received the quickest treatment. In this hypothetical example, interpretability of the model would help us not releasing the flawed model. For instance, one could investigate

a few cases with an interpretability algorithm, and realize that asthma is associated with a decrease in the risk. Such sanity check is not possible "as is" with most models, due to their complexity. We refer to Turner (2016) for other such examples.

The current spread of machine learning in all aspects of our life make the previous example not so hypothetical anymore. Thus there is a pressing need for interpretability. It is interesting to note that this need is recognized by the lawmakers, at least in the European Union, where the European Parliament adopted in 2018 the General Data Protection Regulation (GDPR). Part of the GDPR is a clause on automated decision-making, stating to some extent a right for all individuals to obtain "meaningful explanations of the logic involved." Even if there is an ongoing debate on whether this disposition is binding (Wachter et al., 2017), for the first time, we find written in law the fact that decision-making algorithms cannot stay *black boxes*.

As a response, numerous methods for interpretability were proposed in the recent years. The main question underlying our work is the following:

Do these explanations make sense, in particular when the black-box model f is simple?

In the affirmative, we would like to formally prove this, in order to get certifiable interpretability to some extent. In the negative, we think that this raises concern for the widespread use of such interpretability methods. Indeed, if a particular method fails to explain how a simple linear model predicted a value for a given example, how can we trust this same method to explain how a deep neural network predicted a label for a given image? We believe that there is a need for theoretical guarantees for interpretability. There should be some minimal proof of correctness for any interpretable machine learning algorithm. For instance, showing that one recovers the important coefficients when the model to explain is linear, or that the algorithm is provably discarding coordinates that are not used by the model to explain. This paper attempts to answer these questions in the case of a method called Local Interpretable Model-agnostic Explanations (LIME, Ribeiro et al., 2016). We will see that the answer to both of these questions is *affirmative* when LIME is used for tabular data with default settings.

Without giving too much details on the inner working of LIME (which we will do in Section 2), we want to briefly summarize how it operates. Essentially, to explain an example $\xi \in \mathbb{R}^d$, LIME

(i). creates perturbed examples x_i;

(ii). gets the prediction of the black box model f at this examples;

(iii). weight the predictions with respect to the proximity between x_i and ξ;

(iv). trains a weighted interpretable model on these new examples.

The output of LIME is then the top coefficients of the interpretable model (if it is linear). What makes LIME really powerful and so popular is the use of *interpretable features*, namely discretized features. Instead of saying that "coordinate 3 is important for the prediction $f(\xi)$," LIME indicates to the user that "coordinate 3 being between 1.5 and 7.8 is important." More precisely, LIME outputs a linear, surrogate model, whose coefficients (the *interpretable coefficients*) are given as explanation to the user. These coefficients are the primary center of interest of the present paper.

Contributions. In this paper, we show the following:

- **Explicit expression for LIME's surrogate model.** We show that when the surrogate model is obtained by ordinary least-squares and the number of perturbed samples is large, the interpretable coefficients obtained by Tabular LIME are close to a vector β (with high probability). This is true for any reasonable black-box model f.

- **Role of the hyperparameters.** We give an explicit expression of β, and show several interesting properties thereof. In particular, β is *linear* in f, *stable* with respect to small perturbations of f, and only depend on the bins into which ξ belongs to. We also obtain the behavior of β for small and large bandwidth (the main hyperparameter of the method).

- **Linear model.** When f has some simple algebraic structure, we show how to compute β more explicitly. In particular, when f is linear, we recover the main result of Garreau and von Luxburg (2020), but this time for the default weights, arbitrary bins, and arbitrary input parameters: the explanations provided by Tabular LIME are proportional to the coefficients along each coordinate.

- **Revealing artifacts.** In fact we go past the linear case, showing explicit results for general additive f and multiplicative f. This last case encompass, for instance, indicator functions and radial basis function kernel, for which we demonstrate the accuracy of our predictions. We also leverage the theory to explain precisely artifacts observed when explaining a kernel function with Tabular LIME.

The main difficulty in our analysis comes from the non-linear nature of the new features (defined as indicator functions). In contrast with our previous work Garreau and von Luxburg (2020), we managed to keep the analysis very close to the default implementation (found at `https://github.com/marcotcr/lime` as of August 2020), at the cost of additional notation. As we will see, these additional notation become manageable in the simple cases that we will investigate. The code of all the experiments of the paper can be found at `https://github.com/dgarreau/tabular_lime`.

Related work. LIME is a *posthoc, local* interpretability methods. In other words, it provides explanations (i) "after the fact" (the model is already trained), and (ii) for a specific example. We refer to the exhaustive review papers Guidotti et al. (2018) (especially Section 7.2), and Adadi and Berrada (2018) for an overview of such methods. It seems that LIME has quickly become one of the most widely-used posthoc interpretability methods. But besides the practical interest, LIME has also generated consequent academic attention, with many variations on the method being proposed to specific settings in the last 3 years. For instance the same authors later proposed Anchor-LIME (Ribeiro et al., 2018), also based on the production of perturbed examples, but producing simpler "if-then" rules as explanations. Further specializations of LIME were proposed, in the context of time series analysis (Mishra et al., 2017), and survival model analysis (Kovalev et al., 2020; Utkin et al., 2020).

However, few papers looked into the theoretical analysis of the method. A notable exception is Lundberg and Lee (2017), which considers a generalization of LIME called

SHAP (SHapley Additive exPlanations). In this paper, three desirable properties for an additive feature interpretability method are considered: (i) local accuracy (matching the predictions of the global model locally), (ii) missingness (missing features get a zero weight), and (iii) consistency. It is shown that the only possibility to satisfy the three properties is given by SHAP with a specific choice of weights. However, these weights coming from game theory (Shapley, 1953) are hard to compute. It is nevertheless a beautiful result, and Corollary 1 is very close in spirit to Corollary 14 below—though for another algorithm.

The closest work to the present paper is our previous work Garreau and von Luxburg (2020), which considers a modification of the LIME algorithm for tabular data. Namely, the interpretable components are chosen in a very specific way (the quantiles are those of a Gaussian distribution), and the parameters of the algorithm match the mean of this Gaussian. Moreover, the weights of the perturbed examples are not the weights used in the default implementation of LIME. In this setting, we showed that the values of the interpretable coefficients stabilize towards some values which are attained in closed-form when the model to interpret is *linear*. We explain in the Appendix how the present paper can recover this analysis. Leveraging these closed-form expression, we also showed that cancellation of the interpretable coefficients can happen for some choice of the hyperparameters of LIME. We also demonstrate this effect in Section 4.1 of the present paper, this time for default settings. Finally, we extended the analysis much further. In particular, we consider (i) arbitrary discretization, (ii) general weights including the weights chosen by the default implementation, and (iii) non-linear models, including radial basis kernel function and indicator functions.

Summary of the paper. In Section 2, we introduce Tabular LIME. Section 3 contains our main result, as well as a short discussion and an outline of the proof. The implications of the result for more explicit models are discussed further in Section 4. All the proofs and additional results are collected in Appendix.

2. Tabular LIME

In this section we present Tabular LIME and fix the notation of the paper while doing so. Originally proposed for text and image data, the public implementation of LIME also has a version for tabular data (*i.e.*, data lying in $\mathbb{R}^d$). It is on this version that we focus in the present paper.

We will assume that all the features are *continuous*. Indeed, when there are some discrete features, the sampling scheme of Tabular LIME changes slightly and so would our analysis. Note that the default implementation discretizes the continuous features: this is the road we will follow.

2.1 Quick overview of the algorithm

Regression. The core procedure of LIME is designed for regression: LIME aims to explain a real-valued model $f : \mathbb{R}^d \to \mathbb{R}$ at a fixed example $\xi \in \mathbb{R}^d$. If one desires to apply LIME for classification, then one must use LIME for regression with f the likelihood of being in a given class, that is,

$$f(x) = \mathbb{P}\left(y(x) = c | \zeta_1, \dots, \zeta_m\right) ,$$

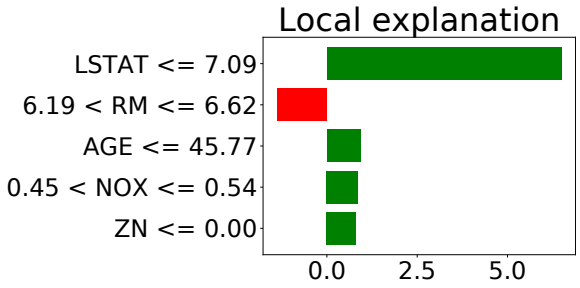

Figure 1: Example output of Tabular LIME. Here we considered a random forest classifier on the Boston Housing dataset (Harrison Jr. and Rubinfeld, 1978). Each row corresponds to a bin along a particular feature, the bar in each row shows the influence of this discretized feature for the prediction according to Tabular LIME. This influence can be positive (green) or negative (red). Here we can see (for instance) that a low value of LSTAT (lower status population) had a positive influence on the predicted variable, the price. See Figure 4 for an alternative presentation.

where $\zeta_1, \ldots, \zeta_m \in \mathbb{R}^d$ is a training set. In this paper, we study Tabular LIME in the context of *regression*, with the undertone that it is possible to transpose our results for *classification*.

Algorithm 1 Getting summary statistics from training data

Require: Train set $\mathcal{X} = \{\zeta_1, \ldots, \zeta_m\}$, number of bins p

for $j = 1$ to d **do**
  for $b = 0$ to p **do**
    $q_{j,b} \leftarrow \texttt{Quantile}(\zeta_{1,j}, \ldots, \zeta_{m,j}; b/p)$ ▷ split each dimension into p bins
  for $b = 1$ to p **do**
    $\mathcal{S} \leftarrow \{\zeta_{i,j} \quad \text{s.t.} \quad q_{j,b-1} < \zeta_{i,j} \leq q_{j,b}\}$ ▷ get the training data falling into bin b
    $\mu_{j,b} \leftarrow \texttt{Mean}(\mathcal{S})$ ▷ compute the empirical mean
    $\sigma^2_{j,b} \leftarrow \texttt{Var}(\mathcal{S})$ ▷ compute the empirical variance
return $q_{j,b}, \mu_{j,b}$, and $\sigma^2_{j,b}$

General overview. Let us now detail how Tabular LIME operates, a prerequisite to the analysis we aim to conduct. We begin with a high-level description of the algorithm. In the next section, we detail each step and fix additional notation.

– **Step 1: binning.** First, Tabular LIME creates interpretable features by splitting each feature's input space in a fixed number of bins p. The idea is to have ranges, not only features, as outputs of the algorithm. As in Figure 1, we prefer to know that a low value of a parameter is important for the prediction, not only that the parameter itself is important. We explain the bin creation in more details in the next section and we refer to Algorithm 1.

– **Step 2: sampling.** Second, Tabular LIME samples perturbed examples $x_1, \ldots, x_n$. For each new example, Tabular LIME samples a bin uniformly at random on each axis and then samples according to a truncated Gaussian whose parameters are given in input to the algorithm. When no training data is provided, one can also directly provide to Tabular LIME the coordinates of the bins and the mean and variance parameters for the sampling. The intuition is to try and mimic the distribution of the data, even though this data may not be readily accessible (remember that LIME aims to explain a black-box model f). We describe the sampling procedure in more details in the next section and we refer to Algorithm 2 for a synthetic view.

– **Step 3: surrogate model.** Finally, a surrogate model is trained on the interpretable features, weighted by some positive weights π_i depending on the distance between the x_is and ξ. The final product is a visualization of the most important coefficient if no feature selection mode is selected by the user (which is the default mode of the algorithm if $d \geq 6$). Algorithm 3 summarizes Tabular LIME, while Figure 1 presents a typical output.

Algorithm 2 `Sample`: Sampling a perturbed example

Require: Bin boundaries $q_{j,p}$, mean parameters $\mu_{j,p}$, variance parameters $\sigma^2_{j,p}$, bin indices of the example to explain, $b^\star$

for $j = 1$ to d **do**
  $b_j \leftarrow \texttt{SampleUniform}(\{1, \ldots, p\})$ ▷ sample bin index
  $(q_\ell, q_u) \leftarrow (q_{j,b_j-1}, q_{j,b_j})$ ▷ get the bin boundaries
  $x_j \leftarrow \texttt{SampleTruncGaussian}(q_\ell, q_u, \mu_{j,b}, \sigma^2_{j,b})$ ▷ sample a truncated Gaussian
  $z_{i,j} \leftarrow \mathbb{1}_{b_j = b^\star_j}$ ▷ mark one if same box
return x,z

In order to analyze Tabular LIME, we need to be more precise with regards to the operation of the algorithm. We now proceed to give more details about the sampling procedure (Section 2.2) and the training of the surrogate model (Section 2.3).

Algorithm 3 Tabular LIME for regression, default implementation

Require: Black-box model f, number of new samples n, example to explain ξ, positive bandwidth ν, number of bins p, bin boundaries $q_{j,b}$, means $\mu_{j,b}$, variances $\sigma^2_{j,b}$

$b^\star \leftarrow \texttt{BinIDs}(\xi, q)$ ▷ get the bin indices of ξ
for $i = 1$ to n **do**
  $x_i, z_i \leftarrow \texttt{Sample}(q, \mu, \sigma, b^\star)$
  $\pi_i \leftarrow \exp\left(\frac{-\|\mathbb{1} - z_i\|^2}{2\nu^2}\right)$ ▷ compute the weight
$\hat{\beta} \in \arg\min_{\beta \in \mathbb{R}^{d+1}} \sum_{i=1}^n \pi_i (f(x_i) - \beta^\top z_i)^2 + \Omega(\beta)$ ▷ compute the surrogate model
return top coefficients of $\hat{\beta}$

2.2 Sampling perturbed examples

In this section, we explain exactly how the sampling of perturbed examples introduced in Algorithm 2 operates.

Defining the bins. The first step in Tabular LIME is to create *interpretable features* along each dimension $j \in \{1, \dots, d\}$. This is achieved by splitting each dimension in $p \geq 2$ bins: the interpretable feature along dimension j is then "belonging to bin p." The intuition for this is added interpretability: instead of knowing that a coordinate is important for the prediction, Tabular LIME gives a range of values for the coordinate that is important for the prediction. We emphasize that p is constant across all dimensions $1 \leq j \leq d$. This is the default behavior of Tabular LIME, although it can happen that $p_j < p$ if there are not enough data along an axis. We will assume throughout the paper that it is never the case, even though the current analysis can be extended to p_j varying across dimensions. Note that if $p = 1$, there are no bins: $z_{ij} = 1$ for any i, j and the surrogate model is just the (weighted) empirical mean.

The boundaries of the bins are an input of Tabular LIME (Algorithm 3). For each feature $j \in \{1, \dots, d\}$, we denote these boundaries by

$$q_{j,0} < q_{j,1} < \cdots < q_{j,p}\,.$$

In addition to the bins boundaries, Tabular LIME takes as input some mean and variance parameters for each bin along each dimension. We denotes these by $(\mu_{j,b}, \sigma^2_{j,b}) \in \mathbb{R} \times \mathbb{R}_+$. As we will see in the next paragraph, these parameters (bin boundaries, means, and standard deviation) are computed from a training set if provided.

Algorithm 4 `BinID`: Getting the bin indices of the instance to explain

Require: Example to explain ξ, bin boundaries $q_{j,b}$

```
for j = 1 to d do
    for b = 1 to p do
        if q_{j,b-1} < ξ_j < q_{j,b} then
            b*_j = b
            break
return b*
```

Training data. In the default operation mode of Tabular LIME, a training set

$$\mathcal{X} = \{\zeta_1, \dots, \zeta_m\}$$

is given as an input to Tabular LIME. Note that this training set is *not necessarily* the training set used to train the black-box model f. In that case, the boundaries of the bins are obtained by considering the quantiles of $\mathcal{X}$ across each dimension. Intuitively, along each dimension j, we split the real line in p bins such that a proportion $1/p$ of the data falls into each bin along this axis. More formally, the boundaries are obtained by taking the quantiles of level b/p for $b \in \{0, 1, \dots, p\}$. That is, the $q_{j,b}$ are such that

$$\forall j, b, \quad \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}_{\zeta_{i,j} \in [q_{j,b-1}, q_{j,b}]} = \frac{1}{p}\,.$$

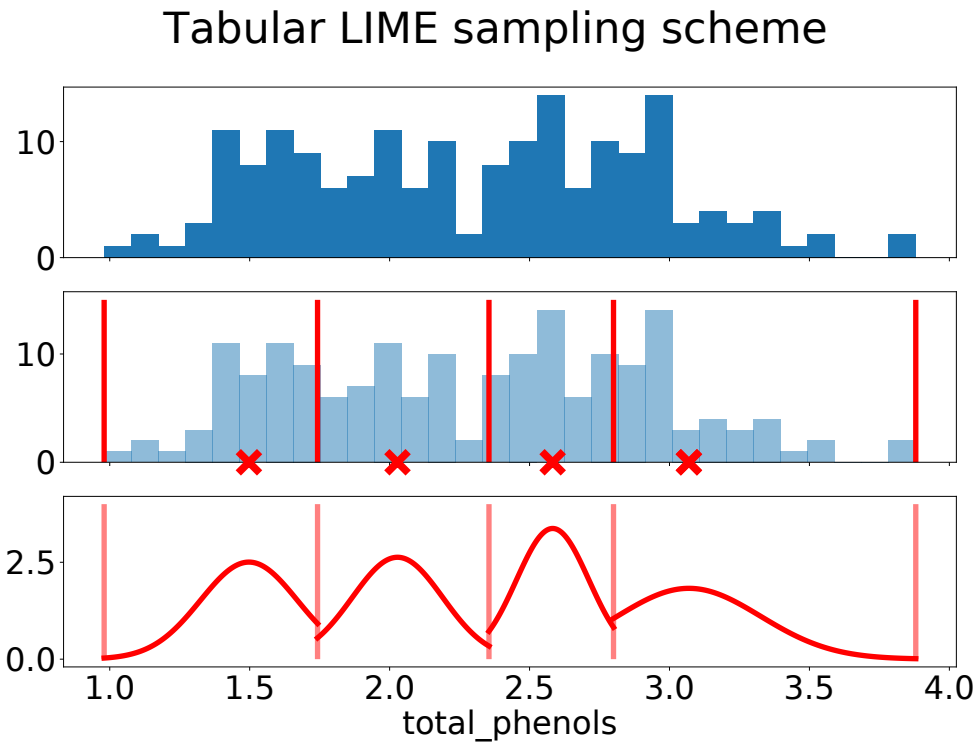

Figure 2: In this figure, we demonstrate how Tabular LIME samples perturbed examples. Data come from the Wine dataset (Cortez et al., 1998). *Top panel:* histogram of the values taken by the `total_phenol` parameter in the original dataset. *Middle panel:* Tabular LIME computes the quantiles (here $p = 4$), then the means and standard deviation for each bin (red cross is the mean). *Bottom panel:* choose a bin uniformly at random, then sample according to a truncated Gaussian on the bin with parameters given by the mean and variance on the bin (density in red).

In particular, for each $1 \leq j \leq d$, $q_{j,0} = \min_{1 \leq i \leq m} \zeta_{i,j}$ and $q_{j,p} = \max_{1 \leq i \leq m} \zeta_{i,j}$.

When a training set $\mathcal{X}$ is used, $\mu_{j,b}$ (resp. $\sigma_{j,b}$) corresponds to the mean of the training data on the bin b along dimension j (resp. the standard deviation). We refer to the top panel of Figure 2 for a visual depiction of this process.

Assuming that the example to explain ξ belongs to $\mathcal{S} := \prod_{j=1}^{d} [q_{j,0}, q_{j,p}]$, the support of the training set, then each ξ_j falls into a bin $b_j^\star$ along coordinate j. The (straightforward) computation of $b^\star$ is given by Algorithm 4.

We note that Garreau and von Luxburg (2020) considers the case where $\mu_{j,b} = \mu_j$ and $\sigma_{j,b} = \sigma_j$ for any b. In addition, the bins boundaries $q_{j,b}$ were chosen as Gaussian quantiles. We do not make such assumptions here.

Sampling scheme. The next step is the sampling of n perturbed examples $x_1, \ldots, x_n \in \mathbb{R}^d$. First, along each dimension, Tabular LIME samples the bin indices of the perturbed samples. We write this sample as a matrix $B \in \mathbb{R}^{n \times d}$, where b_{ij} corresponds to the bin

index of example i along dimension j. In the current implementation of Tabular LIME, the b_{ij} are i.i.d. distributed uniformly on $\{1, \dots, p\}$. We denote by b^ξ the bin indices of our specific example ξ: $b^\xi_j = b$ means that ξ falls in the bin with index b along dimension j.

The bin indices $b_{i,j}$s are subsequently used in two ways. On one hand, Tabular LIME creates the binary features based on these bin indices. Formally, we define $z_{i,j} = \mathbb{1}_{b_{i,j}=b^\xi_j}$, that is, we mark one if $b_{i,j}$ is the same bin as the one ξ_j (the jth coordinate of the example ξ) falls into. We collect these binary features in a matrix $Z \in \mathbb{R}^{n\times(d+1)}$ defined as

$$Z := \begin{pmatrix} 1 & z_{11} & z_{12} & \dots & z_{1d} \\ 1 & z_{21} & z_{22} & \dots & z_{2d} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & z_{n1} & z_{n2} & \dots & z_{nd} \end{pmatrix}.$$

On the other hand, the bin indices are used to sample the new examples $x_1, \dots, x_n$. Let $i \in \{1, \dots, n\}$, the new sample x_i is sampled independently dimension by dimension in the following way: $x_{i,j}$ is distributed according to a truncated Gaussian random variable with parameters $q_{j,b_{ij}-1}$, $q_{j,b_{i,j}}$, $\mu_{j,b_{i,j}}$, and $\sigma^2_{j,b_{i,j}}$. More precisely, $x_{i,j}$ conditionally to $b_{i,j} = b$ has a density given by

$$\rho_{j,b}(t) := \frac{1}{\sigma_{j,b}\sqrt{2\pi}} \cdot \frac{\exp\left(\frac{-(t-\mu_{jb})^2}{2\sigma^2_{j,b}}\right)}{\Phi(r_{j,b}) - \Phi(\ell_{j,b})} \mathbb{1}_{t\in[q_{j,b-1},q_{j,b}]}\,, \tag{1}$$

where we set $\ell_{j,b} := \frac{q_{j,b-1}-\mu_{j,b}}{\sigma_{j,b}}$ and $r_{j,b} := \frac{q_{j,b}-\mu_{j,b}}{\sigma_{j,b}}$, and Φ is the cumulative distribution function of a standard Gaussian random variable. We denote by $\mathrm{TN}(\mu, \sigma^2, \ell, r)$ the law of a truncated Gaussian random variable with mean parameter μ, scale parameter σ, left and right boundaries ℓ and r. Note that the means (resp. standard deviations) of these truncated random variables are generally different from the input means (resp. standard deviations). We denote by $\tilde{\mu}_{j,b}$ (resp. $\tilde{\sigma}_{j,b}$) the mean (resp. standard deviation) of a $\mathrm{TN}(\mu_{j,b}, \sigma^2_{j,b}, q_{j,b-1}, q_{j,b})$ random variable. We refer, again, to Figure 2 for an illustration.

It is important to understand that **the sampling of the new examples does not depend on** ξ, but rather on the bin indices of ξ. Therefore, any two given instances to explain will lead to the same sampling scheme provided that they fall into the same bins along each dimension.

2.3 Surrogate model

We now focus on the training of the surrogate model. As announced in Algorithm 3, the new samples receive positive weights given by

$$\pi_i := \exp\left(\frac{-\|\mathbb{1} - z_i\|^2}{2\nu^2}\right), \tag{2}$$

where $\|\cdot\|$ is the Euclidean norm, and ν is a positive bandwidth parameter. Intuitively, this weighting scheme counts in how many coordinates the bin of the perturbed sample differs

from the bin of the example to explain and then applies an exponential scaling. If all the bins are the same (the perturbed sample falls into the same hyperrectangle as ξ), then the weight is 1. On the other hand, if the perturbed sample is "far away" from ξ, π_i can be quite small (depending on ν). In the default implementation of Tabular LIME, the bandwidth parameter ν is fixed to $\sqrt{0.75d}$. Our main result is actually true for more general weights, see Appendix A for their definition. In particular, this generalization includes the weights studied in Garreau and von Luxburg (2020) as a special case. We collect the weights in a diagonal matrix $W \in \mathbb{R}^{n \times n}$ given by

$$W := \begin{pmatrix} \pi_1 & 0 & \dots & 0 \\ 0 & \pi_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \pi_n \end{pmatrix}.$$

The local surrogate model of LIME is then obtained by optimizing a regularized, weighted square loss

$$\hat{\beta}_n \in \underset{\beta \in \mathbb{R}^{d+1}}{\arg\min} \left\{ \sum_{i=1}^{n} \pi_i (f(x_i) - \beta^\top z_i)^2 + \Omega(\beta) \right\}, \tag{3}$$

where z_i is the ith row of Z. The coefficients of the surrogate model, collected in $\hat{\beta}_n$, are the central output of Tabular LIME and will be our main focus of interest. We often refer to the coordinates of $\hat{\beta}_n$ as the *interpretable coefficients*.

In the default implementation, $\Omega(\beta) = \|\beta\|^2$, where $\|\cdot\|$ is the Euclidean norm. That is, Tabular LIME is using ridge regression. But the default choice of hyperparameters makes the surrogate models obtained by ridge in this setting indistinguishable from *ordinary least-squares*. More precisely, Tabular LIME is using the `scikit-learn` default implementation of ridge, which has a penalty constant equal to one (that is, $\Omega(\beta) = \|\beta\|^2$). Since the weights π_i belong to $[0, 1]$, the $\|y - ZW\beta\|^2$ term in Eq. (3) dominates unless the penalty constant is at least of order n. But the default n is 500 and we investigate the limit in large sample size (meaning that n will often be chosen to be larger than 500), due to the parameter settings of the default implementation, there is virtually no difference between taking ordinary least-squares and ridge. Therefore, even though our analysis is true only for ordinary least-squares ($\Omega = 0$), it recovers the results of the default implementation as soon as n is reasonably large. As a consequence, all the experiments in the paper are done with the default parameters. We demonstrate this phenomenon in Figure 3.

Feature selection. In the final step of Tabular LIME, the user is presented with a visualization of the largest coefficients of the surrogate model. Note that some feature selection mode can be used before this last step. That is, the final output of Tabular LIME is not the given of all coefficients of the surrogate model. However, by default, when the dimension is greater than 6, no feature selection is used and the user is presented with the top 5 interpretable coefficients, as in Figure 1. Therefore we do not consider feature selection in the present work. We will report all the interpretable coefficients since there is randomness in the ranking of the coefficients due to the randomness of the sampling. Note also that because of this randomness in the construction of $\hat{\beta}$ *via* the sampling of the

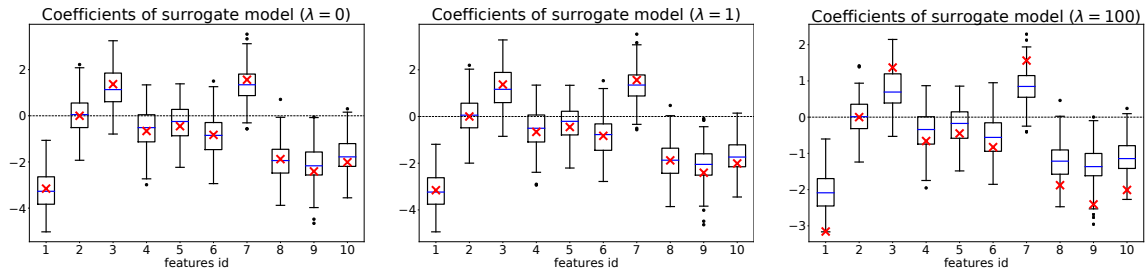

Figure 3: Effect of the regularization on the surrogate model. The black box model is linear. We report 100 runs of Tabular LIME for 1000 perturbed samples on the same ξ. In red, the theoretical predictions given by Theorem 1. *Left panel:* no regularization, the surrogate model is trained with ordinary least-squares. This situation corresponds to our analysis. *Middle panel:* $\lambda = 1$, default choice in `scikit-learn`. We take this default choice of regularization in all our experiments. *Right panel:* $\lambda = 100$. When λ is of order n, the number of perturbed samples, one begins to see the effect of regularization. In effect, the interpretable coefficients are shrinked, and our theoretical predictions only yields an upper envelope.

perturbed examples $x_1, \ldots, x_n$, we will always report the result of several runs of Tabular LIME on any given example, see Figure 4.

Let us summarize the implementation choices for Tabular LIME that we consider in our analysis. The d features are considered to be continuous and are discretized along p bins (the option `discretize_continuous` is set to true, which is default). The default choice is $p = 4$ (`discretizer='quartile'` is default), however we will sometimes use another value for p and leave this parameter free. The bin boundaries $q_{j,b}$ as well as the location and scale parameter for the sampling $\mu_{j,b}$ and $\sigma_{j,b}$ are arbitrary (computed from the appropriate dataset unless otherwise mentioned). We consider default weights given by Eq. (9), with prescribed bandwidth ν. Therefore the critical hyperparameters are p and μ, and we will focus mostly on them.

3. Main result: explicit expression for $\hat{\beta}_n$ in the large sample size

In this section, we state our main result. In a nutshell, when the number of new samples n is large, $\hat{\beta}_n$ (the random vector containing the coefficients of the surrogate model given by Tabular LIME) concentrates around a vector β, for which we provide an explicit expression. This explicit expression depends on f, the model to explain, as well as the parameter of Tabular LIME introduced in the previous section. We state the result in Section 3.1 and present some immediate consequences in Section 3.2. We then present a brief outline of the proof in Section 3.3.

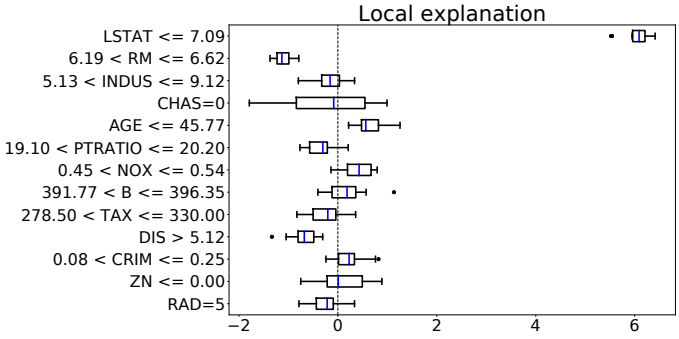

Figure 4: Example output of Tabular LIME, 100 repetitions. Since there is randomness in the sampling of the perturbed samples, we run Tabular LIME several time on the same instance and report the whisker boxes associated to these repetitions. In blue the mean over all repetitions. The vertical dotted line marks zero. We also report all the interpretable component values rather than just the top five. This presentation will be standard for the remainder of the article, although we will often omit the bin boundaries when considering toy data.

3.1 Explicit expression for $\hat{\beta}_n$ in the large sample size

In order to make our result precise, in particular to define the vector β, we need to introduce further notation.

Recall that p is the fixed number of bins along each dimension and $\nu > 0$ is the bandwidth parameter for the weights are the main free parameters of Tabular LIME. A key normalization constant in our computations is given by

$$c := \frac{1}{p} + \left(1 - \frac{1}{p}\right) \mathrm{e}^{\frac{-1}{2\nu^2}} \ . \tag{4}$$

We will also denote by $x \in \mathbb{R}$ a random variable that has the same law as the x_is (recall that they are i.i.d. random variables by construction). To this x is associated a vector of bin indices $b \in \{1, \ldots, p\}^d$, binary features $z \in \{0,1\}^d$ and a weight $\pi \in \mathbb{R}_+$ in the same way x_i was associated to b_i, z_i, and π_i. All expectations in the following are taken with respect to x.

We are now armed with enough notation to state our main result.

Theorem 1 (Explicit expression for $\hat{\beta}_n$ in the large sample size). *Assume that Tabular LIME operates with the default weights (Eq. (2)) and fits the surrogate model with ordinary least squares ($\Omega = 0$ in Eq. (3)). Set $\varepsilon > 0$ and $\eta \in [0,1]$. Suppose that the function to explain f is bounded by a positive constant M on $\mathcal{S} = \prod_j [q_{j,0}, q_{j,p}]$. For any example $\xi \in \mathcal{S}$ for which we want to create an explanation, define*

$$\beta_0 := c^{-d} \left(1 + \sum_{j=1}^{d} \frac{1}{pc - 1}\right) \mathbb{E}\left[\pi f(x)\right] - c^{-d} \sum_{j=1}^{d} \frac{pc}{pc-1} \mathbb{E}\left[\pi z_j f(x)\right] \ , \tag{5}$$

and, for any $1 \leq j \leq d$,

$$\beta_j := c^{-d} \left(\frac{-pc}{pc-1} \mathbb{E}\left[\pi f(x)\right] + \frac{p^2c^2}{pc-1} \mathbb{E}\left[\pi z_j f(x)\right] \right) . \tag{6}$$

Then, for every

$$n \geq \max \left\{ \frac{4608 M d^2 (d+p)^2 \, \mathrm{e}^{\frac{1}{\nu^2}} \log \frac{8d}{\eta}}{\varepsilon^2 c^{2d}}, \frac{10368 d^3 (p+d)^4 M^2 \, \mathrm{e}^{\frac{2}{\nu^2}} \log \frac{8d}{\eta}}{\varepsilon^2 c^{4d}} \right\} ,$$

we have $\mathbb{P}(\|\hat{\beta}_n - \beta\| \geq \varepsilon) \leq \eta$.

Intuitively, Theorem 1 states that:

– if the number of perturbed samples is large enough, the explanations provided by Tabular LIME for any f at a given example ξ stabilize around a fixed value, β;

– this β has an explicit expression, simple enough that we can hope to use it in order to answer the question we asked in the introduction.

In particular, for reasonably large n, **we can focus on β in order to gain insight on the explanations** provided by Tabular LIME with default settings. This is our agenda in Section 4 by assuming specific structures for f. For the time being, we present some additional consequences of Theorem 1 in Section 3.2, which are true without assuming anything on f other than the boundedness assumption.

It is remarkable that Theorem 1 analyzes Tabular LIME as is, the only difference with the default implementation being $\Omega = 0$. Moreover, Theorem 1 is true under pretty mild assumptions. Essentially, we just require f to be bounded on the bins. If these bins are computed from a training set, S is compact and we are essentially requiring that f be well-defined on S, which is virtually always the case for standard machine learning models. An important assumption that could be overlooked is that ξ should lie in $\mathcal{S} = \prod_j [q_{j,0}, q_{j,p}]$ for the theorem to hold. In particular, Theorem 1 does not say anything about examples to explain that do not belong to the bins provided to the algorithm. We leave the analysis of Tabular LIME explanations for $\xi \notin \mathcal{S}$ for future work.

3.2 Direct consequences: properties of LIME that can be deduced from the explicit expression

We now turn to some immediate consequences of Theorem 1.

3.2.1 Linearity of explanations

We first notice that the vector β **depends linearly on** f. Indeed, let us denote by β^f the vector obtained for a given black-box model f for the time being, to emphasize the dependency in f. A careful reading of Eqs. (5) and (6) reveals that β^f depends on f only through the expectations $\mathbb{E}\left[\pi f(x)\right]$ and $\mathbb{E}\left[\pi z_j f(x)\right]$. Since π and z do not depend on f, by linearity of the expectation, we see that

$$\beta^{f+g} = \beta^f + \beta^g \, .$$

This fundamental remark has two consequences.

First, we will soon specialize Theorem 1 to more explicit models f, in order to answer to our main question. Quite a number of models can be written in an additive form (think of a generalized additive model, a kernel regressor, or a random forest). Linearity will allow us to focus on the *building bricks* of these models.

Second, let us assume that our knowledge of f is imperfect. More precisely, let us split the function to explain in two parts: (i) the part coming from the black-box model f by itself, and (ii) the part coming from small perturbations such as numerical errors or measurement noise. Linearity allows us to focus on the perturbation part separately, and prove the following:

Lemma 2 (Stability of the explanations given by Tabular LIME). *Suppose that $\xi \in \mathcal{S}$. Consider f and g two functions that are bounded on $\mathcal{S}$. Then, under the assumptions of Theorem 1,*

$$\|\beta^f - \beta^g\| \leq \frac{\sqrt{d(9d+4p^2)}\,\mathrm{e}^{\frac{1}{2\nu^2}}}{p-1}\,\|f-g\|_\infty\,.$$

As expected, small perturbations of the function to explain do not perturb the explanations too much, which is a desirable property. Lemma 2 is proven in Appendix E.

3.2.2 Explanations only depend on the bin indices of ξ

The interpretable coefficients β_j depend only on the bin indices $b_j^\star$ of the example ξ. Indeed, Eqs. (5) and (6) reveal that only the sampling of x depends on the actual coordinates of ξ. But if we recall Section 2.2, this sampling only depends on the bin indices of ξ. Therefore, **Tabular LIME provides the same explanation for any two instances falling in the same bins along each dimension**, up to some noise coming from the sampling procedure. See Figure 5 for an illustration of this phenomenon. In a sense, this behavior gives a certain stability to the explanations provided by Tabular LIME: if two examples to explain ξ and ξ' are very close, they are likely to have the same bin indices, and therefore the same β. On the other hand, if ξ and ξ' are close but do not have the same bin indices, the explanations are likely to be quite different. This could be an explanation for the instability of explanations observed by Alvarez-Melis and Jaakkola (2018).

In particular, in the general case, the value of the surrogate model at ξ cannot be the same as $f(\xi)$. **The *local accuracy* property of Lundberg and Lee (2017) is thus not satisfied in the general case by Tabular LIME.**

3.2.3 Dependency on the bandwidth parameter ν

Suppose that the bandwidth is large, that is, $\nu \to +\infty$. In that case, it is clear from the definitions that $c \to 1$ and $\pi_i \to 1$ almost surely. By dominated convergence, Eq. (5) and (6) imply that

$$\beta_0 \longrightarrow \left(1 + \frac{d}{p-1}\right)\mathbb{E}\left[f(x)\right] - \sum_{j=1}^{d}\frac{p}{p-1}\,\mathbb{E}\left[z_j f(x)\right]\,, \tag{7}$$

and, for any $1 \leq j \leq d$,

$$\beta_j \longrightarrow \frac{-p}{p-1}\,\mathbb{E}\left[f(x)\right] + \frac{p^2}{p-1}\,\mathbb{E}\left[z_j f(x)\right]\,. \tag{8}$$

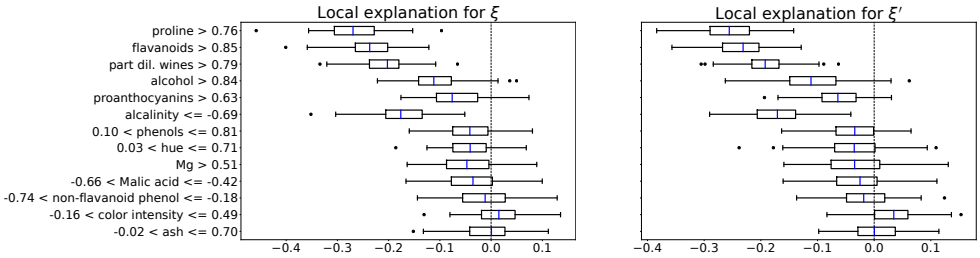

Figure 5: Explanations given by Tabular LIME for a kernel ridge regressor trained on the Wine dataset (Cortez et al., 1998). *Left panel:* Explanations for a given ξ. *Right panel:* We recovered the bins boundaries as well as the bin indices of ξ. We then sampled ξ' in the same d-dimensional box as ξ. The local explanation for ξ' are indiscernible from those of ξ once randomness is taken into account. All parameters are normalized.

We show this convergence phenomenon in Figure 6. In cases where the bandwidth choice of the default implementation $\nu = \sqrt{0.75d}$ is large, this approximation is well-satisfied. In fact, the bandwidth parameter then becomes redundant: it is equivalent to give weight 1 to every perturbed sample.

On the other hand, when $\nu \to 0$, it is straightforward to show that $\beta_j \to 0$ for any $1 \leq j \leq d$. In between these two extremes, the behavior of the interpretable coefficients can be pretty wild. As demonstrated in Figure 6, the interpretable coefficients can even cancel for large, positive values of ν. **This is a worrying phenomenon: for some values of the bandwidth ν, the explanation provided by Tabular LIME is negative, while it becomes positive for other choices.** In the first case, the trusting user of Tabular LIME would grant a positive influence for the parameter, in contrast to a negative influence in the second case. This is not only a theoretical worry: if the values of these interpretable coefficients are large enough, they may be ranked amongst the top five usually displayed to the user using the default settings.[1]

3.3 Outline of the proof of Theorem 1

Before turning to specializations of Theorem 1, we provide a short outline of the proof (the complete proof is provided in appendix).

Since we restrict our analysis to $\Omega = 0$, Eq. (3) becomes a simple weighted least-squares problem, and the solution is given in closed-form by

$$\hat{\beta}_n = \left(Z^\top W Z\right)^{-1} Z^\top W y\,,$$

where we defined $y \in \mathbb{R}^n$ coordinate-wise by $y_i := f(x_i)$. We define $\hat{\Sigma}_n := \frac{1}{n} Z^\top W Z$ and $\hat{\Gamma}_n := \frac{1}{n} Z^\top W y$ and notice that $\hat{\beta}_n = \hat{\Sigma}_n^{-1} \hat{\Gamma}_n$. Elementary computations show that both $\hat{\Sigma}_n$

1. We are not the first to point out the critical role of ν, see for instance `https://christophm.github.io/interpretable-ml-book/lime.html`

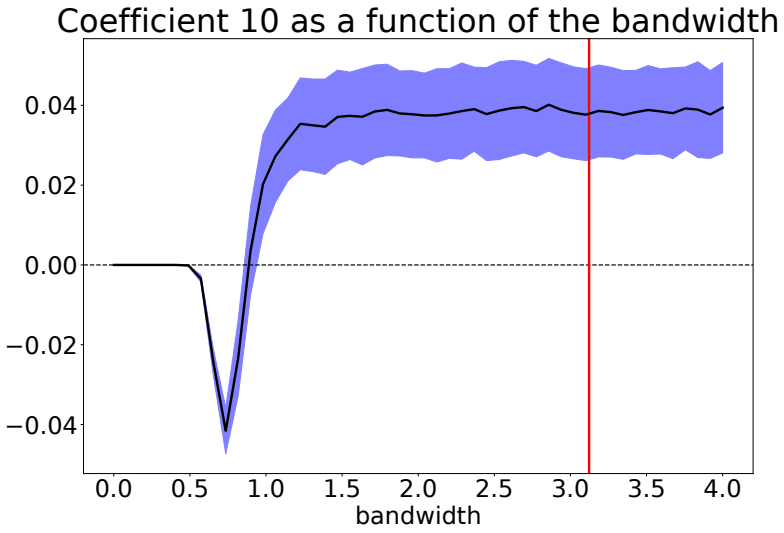

Figure 6: Tabular LIME on a kernel regressor trained on the Wine dataset, 100 repetitions, 2000 samples, plotting the interpretable coefficient for `color_intensity` for varying bandwidth. We see that $\beta_{10} = 0$ when $\nu = 0$ as predicted, while β_{10} stabilizes when $\nu \to +\infty$. We also note that some values of the bandwidth can cancel the interpretable coefficient (here around $\nu = 1$). In red, the default choice for ν, given by $\sqrt{0.75d}$.

and $\hat{\Gamma}_n$ can be written as empirical averages depending on the sampling of the perturbed samples. Since this sampling is i.i.d., the weak law of large numbers guarantees that

$$\hat{\Sigma}_n \xrightarrow{\mathbb{P}} \Sigma \quad \text{and} \quad \hat{\Gamma}_n \xrightarrow{\mathbb{P}} \Gamma\,,$$

where we defined $\Sigma := \mathbb{E}[\hat{\Sigma}_n]$ and $\Gamma := \mathbb{E}[\hat{\Gamma}_n]$. These quantities are given by Lemma 6 and Eq. (22). Since Σ is invertible in closed-form for a large class of weights (Lemma 7), we can set $\beta := \Sigma^{-1}\Gamma$ and proceed to show that $\hat{\beta}_n$ is close from β in probability. A prerequisite to the concentration of $\hat{\beta}_n$ are the concentration of $\hat{\Sigma}_n$ (Lemma 18) and the concentration of $\hat{\Gamma}_n$ (Lemma 20). Together with the control of $\left\|\Sigma^{-1}\right\|_{\text{op}}$ (Lemma 8), a binding lemma (Lemma 23) allows us to put everything together in Appendix H.

4. Special cases of Theorem 1

We now specialize Theorem 1 to three simple models. Namely, we will assume that f is an **additive function** over the coordinates (Section 4.1), **multiplicative** along the coordinates (Section 4.2), and finally a function **depending only on a subset of the coordinates** (Section 4.3). Our goal is to look closer into the explanations provided by Tabular LIME for these simple models.

Before we do that, we need to introduce some additional notation. In order to use Theorem 1 for models with additional structure, we will need to make the expectations computations more explicit. Recall that we set x the random variable corresponding to the sampling of the perturbed examples (see Section 2.2). For any $\psi : \mathbb{R} \to \mathbb{R}$, we now introduce the notation

$$e_{j,b}^{\psi} := \mathbb{E}\left[\mathrm{e}^{\frac{-(1-z_j)^2}{2\nu^2}}\,\psi(x_j)\middle| b_j = b\right]\,. \tag{9}$$

When $\psi = 1$, we just write $e_{j,b}$ instead of $e_{j,b}^1$, and when $\psi = \mathrm{id}$, we write $e_{j,b}^x$.

Even though this looks cumbersome, this notation is going to help a lot. The reason is quite simple: whenever we need to compute an expectation with respect to x, we will use the law of total expectation with respect to the random variables $b_1, \ldots, b_d$, thus effectively conditioning with respect to the events $\{b_j = b\}$ for $b \in \{1, \ldots, p\}$. The idea is to "cut" the expectation depending on which bins x falls into on each dimension.

We also define the normalization constants

$$c_j^{\psi} := \frac{1}{p}\sum_{b=1}^{p} e_{j,b}^{\psi}\,,$$

and $c_j := c_j^1$. Finally, we set $C := \prod_{j=1}^{d} c_j$.

First computations. In the default implementation of Tabular LIME, if $b = b_j^\star$, then $z_j = 1$, and 0 otherwise. Thus the computation of $e_{j,b}$ is straightforward in this case:

$$e_{j,b} = \begin{cases} 1 \text{ if } b = b_j^\star \\ \mathrm{e}^{\frac{-1}{2\nu^2}} \text{ otherwise.} \end{cases}$$

The expression of c_j is also quite simple. In particular, c_j does not depend on j and we find

$$\forall 1 \leq j \leq d, \quad c_j = c := \frac{1}{p} + \left(1 - \frac{1}{p}\right) \mathrm{e}^{\frac{-1}{2\nu^2}},$$

recovering the expression of c given in Section 3. As a consequence, note that $C = c^d$.

We now have all the required tools to specialize Theorem 1 in practical cases.

4.1 General additive models, including linear functions

In this section, we consider functions that can be written as a sum of functions where each function depends only on one coordinate. Namely, we make the following assumption on f:

Assumption 4.1. We say that $f : \mathbb{R}^d \to \mathbb{R}$ is *additive* if there exist arbitrary functions $f_1, \ldots, f_d : \mathbb{R}^d \to \mathbb{R}$ such that, for any $x \in \mathbb{R}^d$,

$$f(x) = \sum_{j=1}^{d} f_j(x_j).$$

General additive models generalize *linear models*, where $f_j(x) = f_j \cdot x$ for any $1 \leq j \leq d$. They were popularized by Stone (1985) and Hastie and Tibshirani (1990). We refer to Chapter 9 in Hastie et al. (2001) for an introduction to general additive models, and in particular Section 9.1.1 regarding the training thereof. If f is a general additive model, we can specialize Theorem 1, and examine the explanation provided by Tabular LIME in this case. Rather than giving (again) the concentration result, we focus directly on β.

Lemma 3 (Computation of β for additive f). *Assume that f satisfies Assumption 4.1 and that the assumptions of Theorem 1 are satisfied (in particular, each f_j is bounded on $[q_{j,0}, q_{j,p}]$). Set $\xi \in \mathcal{S}$. Then Theorem 1 holds with $\beta \in \mathbb{R}^{d+1}$ given by*

$$\beta_0 = \sum_{k=1}^{d} \frac{1}{pc - 1} \sum_{b \neq b_k^\star} e_{k,b}^{f_k},$$

and, for any $1 \leq j \leq d$,

$$\beta_j = \frac{pc}{pc - 1} \left(e_{j,b_j^\star}^{f_j} - \frac{c_j^{f_j}}{c} \right).$$

The proof of Lemma 3 is a direct consequence of Theorem 1 and Lemma 13. Lemma 3 has several interesting consequences.

Splitting the coordinates. A careful reading of the expression of β in Lemma 3 reveals that the j-th interpretable coefficient β_j depends only on f_j, the part of the function depending on the j-th coordinate. In other words, **Tabular LIME splits the explanations coordinate by coordinate for general additive models.** This property is desirable in our opinion. Indeed, since our model f depends on the j-th coordinate only through the function f_j, f_j alone should be involved on the part of the explanation which is concerned by j.

Ignoring unused coordinates. Suppose for a moment that f is additive but does not depend on coordinate j at all. That is, $f_j(x) = \kappa$ for any x, where κ is a constant. Then, by linearity of the conditional expectation, $e^{f_j}_{j,b} = \kappa e_{j,b}$ for any b. By definition of the normalization constant $c^{f_j}_j$, we have $c^{f_j}_j = \kappa c$. Therefore

$$e^{f_j}_{j,b^\star_j} - \frac{c^{f_j}_j}{c} = 0\,,$$

and we deduce immediately that $\beta_j = 0$. In other words, **for an additive f, Tabular LIME provably ignores unused coordinates.** This is also a property that one could reasonably expect from any interpretability algorithm. We show that this property also holds for more general weights in the Appendix (see Lemma 13).

4.1.1 Linear functions

We can be even more precise in the case of linear functions. In this case, the functions f_j are defined as $f_j(x) = f_j \cdot x$, and by linearity we just need to compute $e^x_{j,b}$. Recall that we defined $\tilde{\mu}_{j,b}$ as the mean of the random variable $\mathrm{TN}(\mu_{j,b}, \sigma_{j,b}, q_{j,b-1}, q_{j,b})$. As a consequence, $e^x_{j,b} = \tilde{\mu}_{j,b}$ if $b = b^\star_j$ and $\mathrm{e}^{\frac{-1}{2\nu^2}}\,\tilde{\mu}_{j,b}$ otherwise. We deduce that $c^x_j = \tilde{\mu}_{j,b^\star_j} + \sum_{b\neq b^\star_j} \tilde{\mu}_{j,b}$. Let us set

$$\tilde{\tilde{\mu}}_j = \frac{c^{f_j}_j}{c} = \frac{\tilde{\mu}_{j,b^\star_j} + \sum_{b\neq b^\star_j} \mathrm{e}^{\frac{-1}{2\nu^2}}\,\tilde{\mu}_{j,b}}{1 + (p-1)\,\mathrm{e}^{\frac{-1}{2\nu^2}}}\,.$$

Following Lemma 3, we obtain

$$\beta_0 = f(\tilde{\tilde{\mu}}) - \sum_{j=1}^{d} \frac{1}{pc-1}(\tilde{\mu}_{j,b^\star_j} - \tilde{\tilde{\mu}}_j) f_j\,,$$

and, for any $1 \leq j \leq d$,

$$\beta_j = \frac{pc}{pc-1}(\tilde{\mu}_{j,b^\star_j} - \tilde{\tilde{\mu}}_j) f_j\,.$$

We can simplify further this last display. Indeed,

$$\begin{aligned}
\tilde{\tilde{\mu}}_j - \tilde{\mu}_{j,b^\star_j} &= \frac{\tilde{\mu}_{j,b^\star_j} + \sum_{b\neq b^\star_j} \mathrm{e}^{\frac{-1}{2\nu^2}}\,\tilde{\mu}_{j,b}}{1 + (p-1)\,\mathrm{e}^{\frac{-1}{2\nu^2}}} - \tilde{\mu}_{j,b^\star_j} && \text{(definition of } \tilde{\tilde{\mu}}) \\
&= \frac{\sum_{b\neq b^\star_j} \mathrm{e}^{\frac{-1}{2\nu^2}}\,(\tilde{\mu}_{j,b} - \tilde{\mu}_{j,b^\star_j})}{1 + (p-1)\,\mathrm{e}^{\frac{-1}{2\nu^2}}}\,.
\end{aligned}$$

We deduce that

$$\begin{aligned}
\frac{pc}{pc-1}(\tilde{\mu}_{j,b^\star_j} - \tilde{\tilde{\mu}}_j) &= \frac{1}{p-1}\sum_{b\neq b^\star_j}(\tilde{\mu}_{j,b^\star_j} - \tilde{\mu}_{j,b}) \\
&= \tilde{\mu}_{j,b^\star_j} - \frac{1}{p-1}\sum_{b\neq b^\star_j} \tilde{\mu}_{j,b}\,.
\end{aligned}$$

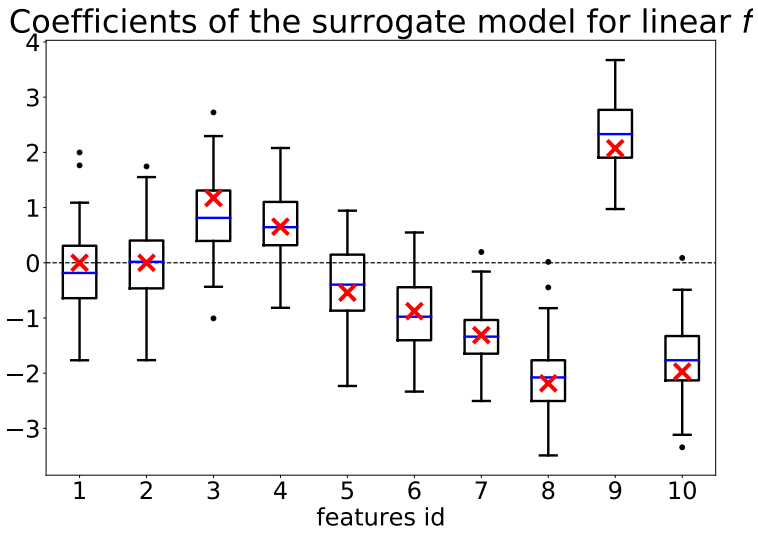

Figure 7: In this set of experiments, we ran Tabular LIME 100 times on a linear f in dimension 10 with bandwidth 1 and 10^3 perturbed samples. The weights are those from the default implementation, the surrogate model found by weighted ordinary least squares. In red, our predictions. In black, the whisker boxes from experimental values. In blue, the experimental mean. We can see that the theoretical predictions of Eq. (10) match the experimental results.

As a consequence,

$$\forall j \geq 1, \quad \beta_j = \frac{f_j}{p-1} \sum_{b=1}^{p} (\tilde{\mu}_{j,b_j^\star} - \tilde{\mu}_{j,b}) \,. \tag{10}$$

We demonstrate in Figure 7 the accuracy of our theoretical predictions.

As in Garreau and von Luxburg (2020), we recover a linear dependency in the f_js: **for a linear model, the interpretable coefficient along dimension j is proportional to the coefficient of the linear model.** This is also, in our opinion, a nice property of any interpretable algorithms: since a linear model is already interpretable to some extent, the interpretable version thereof should coincide up to constants.

However, the proportionality coefficient by which f_j is multiplied depends on the parameters given as input to Tabular LIME. We see that it is thus possible to cancel β_j, leading to the disappearance of the jth interpretable component in the explanation. In particular, a wrong choice of p can achieve this cancellation. We demonstrate this phenomenon in Figure 8. This cancellation is not a good property. One would expect the choice of hyperparameters to not be so brittle.

Let us explain what is happening in Figure 8. We obtained the expression of β_j in Eq. (10). If $\frac{1}{p-1} \sum_{b=1}^{p} (\tilde{\mu}_{j,b_j^\star} - \tilde{\mu}_{j,b}) = 0$, then $\beta_j = 0$, *whatever the value of f_j may be.* We

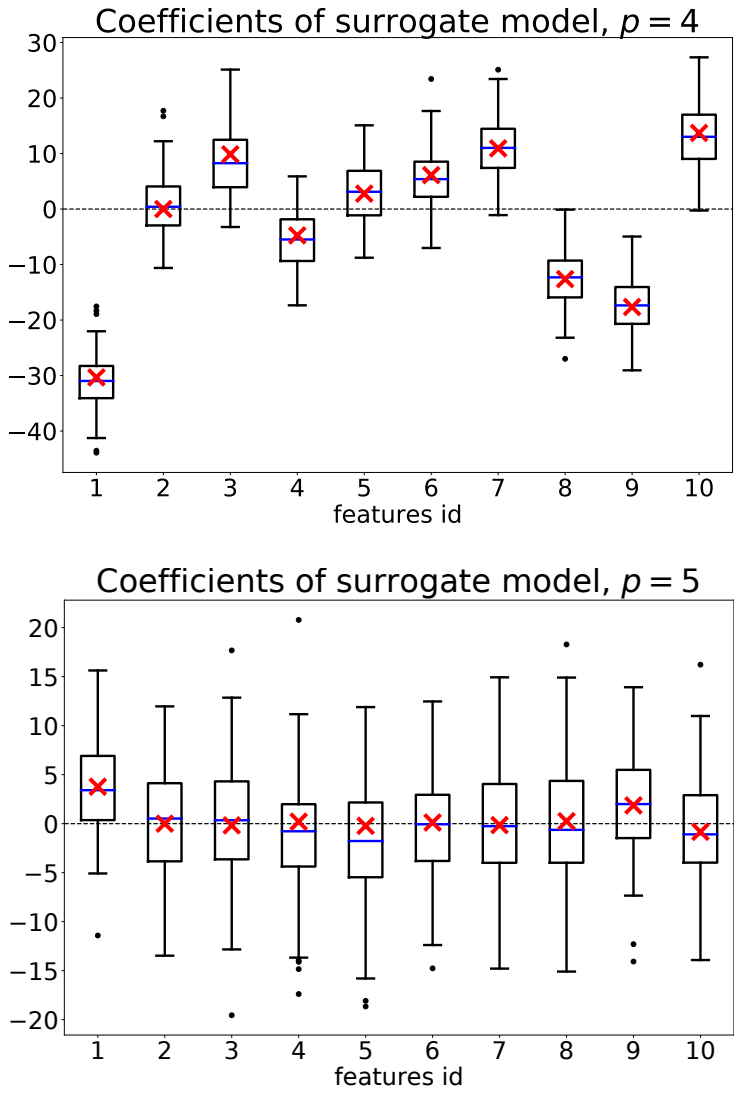

Figure 8: Cancellation phenomenon. Explanation given by Tabular LIME for a linear f in dimension 10, 1000 new samples, 100 experiments. The training set is sampled independently according to a $\mathcal{U}([-10, 10])$ distribution along each coordinate. *Top panel:* $p = 4$ bins. *Bottom panel:* $p = 5$ bins. In red the theoretical values given by Lemma 3, in black the experimental values from Tabular LIME. We can see that, surprisingly, choosing a different number of bins along each dimension (5 instead of 4) sets the values of *all* interpretable coefficients to zero, both in theory and in practice.

can rewrite this condition as

$$\tilde{\mu}_{j,b_j^\star} = \frac{1}{p-1} \sum_{b \neq b_j^\star} \tilde{\mu}_{j,b} \,.$$

Intuitively, along dimension j, if the means on the boxes balance the mean on the special box, then β_j vanishes. In the experiment depicted in Figure 8, we considered a uniformly distributed training set on $[-B, B]$, with B a positive constant. Then the means $\mu_{j,b}$ are evenly distributed across $[-B, B]$, as well as the modified means $\tilde{\mu}_{j,b}$. Thus, if we consider a ξ in a central position, the following happens:

- if p is even, then $\tilde{\mu}_{j,b_j^\star}$ and $\frac{1}{p-1}\sum_{b \neq b_j^\star} \tilde{\mu}_{j,b}$ are far away (top panel of Figure 8);
- if p is odd, then $\tilde{\mu}_{j,b_j^\star}$ is in a central position and approximately equal to $\frac{1}{p-1}\sum_{b \neq b_j^\star} \tilde{\mu}_{j,b}$. The corresponding coefficient vanishes (bottom panel of Figure 8).

It is interesting to note, however, that it is not possible to cancel out the interpretable coefficient by a clever choice of bandwidth in Eq.(10), contrarily to what is done in Garreau and von Luxburg (2020). In fact, the magnitude of the explanations does not depend on the bandwidth at all. This seems due to the specific setting considered in Garreau and von Luxburg (2020), especially choosing $\mu_{j,b}$ and $\sigma_{j,b}$ not depending on b.

4.2 Multiplicative models, including indicator functions and Gaussian kernels

In this section, we turn to the study of functions f that can be written as a product of functions where each term depends only on one coordinate. Namely, we now make the following assumption on f:

Assumption 4.2. We say that f is *multiplicative* if there exist functions $f_1, \dots, f_d : \mathbb{R} \to \mathbb{R}$ such that

$$\forall x \in \mathbb{R}^d, \quad f(x) = \prod_{j=1}^{d} f_j(x_j) \,.$$

In this case, as promised, we also can be more explicit in the statement of Theorem 1. As before, we only report the value of β, since the concentration result remains unchanged.

Lemma 4 (Computation of β, multiplicative f). *Assume that f satisfies Assumption 4.2 and suppose that the assumptions of Theorem 1 holds (in particular, for each $1 \leq j \leq d$, f_j is bounded on $[q_{j,0}, q_{j,p}]$). Set $\xi \in \mathcal{S}$. Then Theorem 1 holds with $\beta \in \mathbb{R}^{d+1}$ given by*

$$\beta_0 = \frac{\prod_{k=1}^{d} c_k^{f_k}}{C} \left[1 + \sum_{j=1}^{d} \frac{1}{pc-1} \left(1 - e_{j,b_j^\star}^{f_j} \cdot \frac{c}{c_j^{f_j}} \right) \right] ,$$

and, for any $1 \leq j \leq d$,

$$\beta_j = \frac{\prod_{k=1}^{d} c_k^{f_k}}{C} \cdot \frac{pc}{pc-1} \left(e_{j,b_j^\star}^{f_j} \cdot \frac{c}{c_j^{f_j}} - 1 \right) .$$

Lemma 4 is a consequence of Lemma 15 in the Appendix (which is true for general weights).

As in the additive case, we can see that unused coordinates are forgotten in the explanation, a desirable property. Indeed, let us assume that for a certain index j, the function f does not depend on the jth coordinate. In other words, $f_j(x) = \kappa$ for any $x \in \mathbb{R}$. Then, $e^{f_j}_{j,b^\star_j} = \kappa e_{j,b^\star_j}$ and $c^{f_j}_j = \kappa c_j$. It follows that $\beta_j = 0$. For a multiplicative f, Tabular LIME provably ignores unused coordinates. This is also true for arbitrary weights (see Lemma 15 in the Appendix).

We now give two fundamental examples in which multiplicative functions are the building block of the model: tree-based regressors and kernel-based regressors.

4.2.1 Tree-based methods

Tree-based methods partition the feature space and then fit a simple model on each element of the partition (see Section 9.2 in Hastie et al. (2001) for an introduction). They are the building brick of very popular regression algorithms such as random forests (Breiman, 2001), which are considered as one of the most successful general-purpose algorithms in modern-times (Biau and Scornet, 2016). Do the explanations provided by Tabular LIME make sense when f is a tree-based regressor? In this section, we will consider one of the most popular tree-based method, CART (Breiman et al., 1984). Without getting into too much details about the construction of these trees, we can describe their construction in two simple steps: (i) create a partition of the input space $A_1 \cup \cdots \cup A_N$, where each A_i is an hyper-rectangle with faces parallel to the axes, and (ii) fit a simple model on each A_i, for instance a constant equal to the mean of the training set on the hyper-rectangle A_i. Simply put, once trained, CART output predictions according to

$$f(x) = \sum_{i=1}^{N} \alpha_i \mathbb{1}_{x \in A_i} \,,$$

where $\alpha_i \in \mathbb{R}$ for each $1 \leq i \leq N$ and $A_i = \prod_{j=1}^{d} [s_j, t_j]$ are hyper-rectangles. By linearity, if we want to understand how Tabular LIME produces interpretable coefficients for f, we only need to understand how Tabular LIME produces interpretable coefficients for the function $\mathbb{1}_A : x \mapsto \mathbb{1}_{x \in A}$ for a given hyper-rectangle $A \subseteq \mathbb{R}^d$. The explanation for f will just be the (weighted) sum of the explanations.

We now make the fundamental observation that $\mathbb{1}_A$ is a *multiplicative* function. Indeed, we can write $\mathbb{1}_{x \in A} = \prod_{j=1}^{d} a_j(x_j)$, with

$$a_j(x) = \mathbb{1}_{x \in [s_j, t_j]} \,,$$

where $s_j < t_j$ are the boundaries of the hyper-rectangle A along dimension j. Hence we can apply Lemma 4 to the model $\mathbb{1}_A$.

Let us see how to compute the interpretable coefficients in this case. Our first task is to compute the $e^{a_j}_{j,b}$ coefficients. By definition of the $e_{j,b}$, we have

$$e^{f_j}_{j,b} = \mathrm{e}^{\frac{-\mathbb{1}_{b=b^\star_j}}{2\nu^2}} \, \mathbb{E}\left[\mathbb{1}_{x_j \in [s_j, t_j]} \middle| b_j = b \right] =: \mathrm{e}^{\frac{-\mathbb{1}_{b=b^\star_j}}{2\nu^2}} \, e^t_{j,b} \,. \tag{11}$$

Since x_j has support on $[q_{j,b-1}, q_{j,b}]$ conditionally to $b_j = b$, it is straightforward to compute $e^t_{j,b}$ with respect to the relative position of $[s_j, t_j]$ and $[q_{j,b-1}, q_{j,b}]$. In particular, $e^t_{j,b} = 0$ if the intersection is empty, and we find

$$e^t_{j,b} = \frac{\Phi\left(\frac{t_j \wedge q_{j,b} - \mu_{j,b}}{\sigma_{j,b}}\right) - \Phi\left(\frac{s_j \vee q_{j,b-1} - \mu_{j,b}}{\sigma_{j,b}}\right)}{\Phi\left(\frac{q_{j,b} - \mu_{j,b}}{\sigma_{j,b}}\right) - \Phi\left(\frac{q_{j,b-1} - \mu_{j,b}}{\sigma_{j,b}}\right)} \tag{12}$$

otherwise. We deduce the value of the normalization constants $c_j^{f_j}$, and, as a consequence of Lemma 4, a closed-form expression for the interpretable coefficients of $f(x) = \mathbb{1}_{x \in A}$. We illustrate in Figure 9 the accuracy of these prediction.

Looking closely at the expression of the β_js given by Lemma 4, we see that the value of β_j for a fixed j is large if

$$e^{f_j}_{j,b^\star_j} \gg \frac{c_j^{f_j}}{c} = \frac{e_{j,b^\star_j} + \sum_{b \neq b^\star_j} e_{j,b}}{1 + (p-1)\,\mathrm{e}^{\frac{-1}{2\nu^2}}}\,,$$

since the multiplicative constants do not depend on j. Using the expression of the $e_{j,b}$ as a function of the $e^t_{j,b}$, we deduce that β_j is large if

$$e^t_{j,b^\star_j} \gg \frac{1}{p-1} \sum_{b \neq b^\star_j} e^t_{j,b}\,.$$

In other words, an interpretable coefficient is large if, on this coordinate, the coefficient $e^t_{j,b^\star_j}$ is larger than the typical $e^t_{j,b}$. But recall that $e^t_{j,b} = 0$ whenever $[s_j, t_j] \cap [q_{j,b-1}, q_{j,b}] = \emptyset$. Therefore, β_j takes a high value (resp. low) when $\xi_j \in [s_j, t_j]$ (resp. $\notin$), modulo side effects due to the bins. In a sense, β_j behaves as a *detector* of $[s_j, t_j]$: β_j is large if the bin $[q_{j,b^\star_j - 1}, q_{j,b^\star_j}]$ containing ξ_j intersects $[s_j, t_j]$. Interestingly, this does not depend on the bandwidth parameter ν.

As a direct consequence, we can see that the interpretable coefficients are small when ξ is far away from A. This is a desirable property: we want the explanations to be zero since the function is completely flat in this case. Note however that there is a notable exception: whenever ξ_j is close to $[s_j, t_j]$, then β_j is no longer small. See Figure 10 for an illustration of this phenomenon. We believe that this is not a desirable behavior, since these artifacts will be added up when considering the explanation for the total partition, possibly creating wrong explanations.

4.2.2 Radial basis function kernel

An important class of examples is given by kernel regressors, a family of predictors containing, for instance, kernel support vector machines (see Hastie et al. (2001) Section 12.3 and Schoelkopf and Smola (2001)). In all these cases, f, the model to explain can be written in the form

$$f(x) = \sum_{i=1}^{m} \alpha_i k(x, \zeta_i)\,,$$

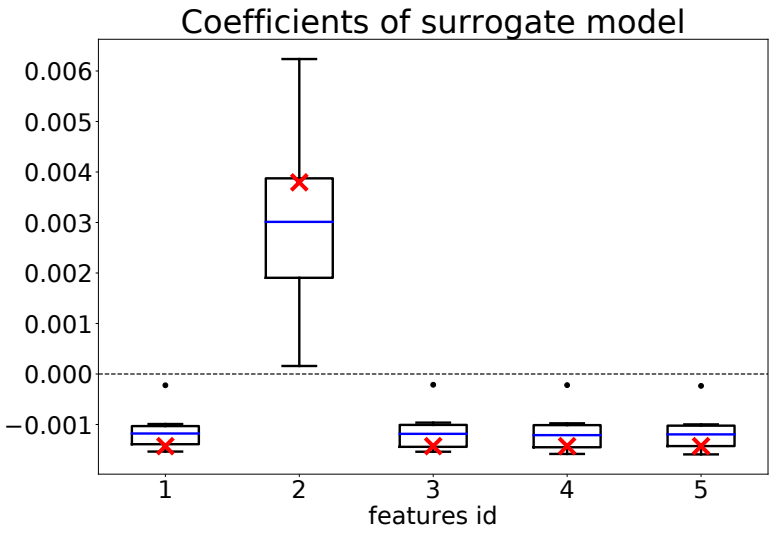

Figure 9: Theory vs practice for an indicator function. The empirical values were obtained by running Tabular LIME with 10^4 samples. We repeated the experiment 10 times. In red, the theoretical values given by Lemma 4 in conjunction with Eq. (11). The slightly negative values correspond to coordinates where ξ_j is not aligned with A, whereas the positive value does.

where k is a positive semi-definite kernel, $\alpha \in \mathbb{R}^m$ are some real coefficients, and the ζ_is are support points. By linearity of the explanation, we can focus on the β corresponding to $k(\cdot, \zeta_i)$. For one of the most intensively used kernel, the *radial basis function* kernel (also called Gaussian kernel), $x \mapsto k(x, \zeta_i)$ is *multiplicative*. Therefore, we can compute in closed-form the β associated to these models. We detail the computation of β in this case.

Let us set $f(x) = \exp\left(\frac{-\|x-\zeta\|^2}{2\gamma^2}\right)$ for some positive γ and a fixed $\zeta \in \mathbb{R}^d$. We see that f is multiplicative, with $f_j(x) = \exp\left(\frac{-(x-\zeta_j)^2}{2\gamma^2}\right)$. Therefore we can use Proposition 4 to compute the associated interpretable coefficients. Computing $e_{j,b}$ requires to integrate f_j with respect to the Gaussian measure. We can split the square as

$$\frac{(x-\mu_{j,b})^2}{2\sigma_{j,b}^2} + \frac{(x-\zeta_j)^2}{2\gamma^2} = \frac{(x-\tilde{m}_{j,b})^2}{2\tilde{s}_{j,b}^2}\,,$$

where we set

$$\tilde{m}_{j,b} := \frac{\gamma^2\mu_{j,b} + \sigma_{j,b}^2\zeta}{\gamma^2 + \sigma_{j,b}^2}\,, \quad \text{and} \quad \tilde{s}_{j,b}^2 := \frac{\sigma_{j,b}^2\gamma^2}{\sigma_{j,b}^2 + \gamma^2}\,.$$

Let $x \sim \mathrm{TN}(\mu_{j,b}, \sigma_{j,b}^2, q_{j,b-1}, q_{j,b})$. We deduce that

$$\mathbb{E}\left[\exp\left(\frac{-(x-\zeta_j)^2}{2\gamma^2}\right)\right] = \frac{\tilde{s}_{j,b}}{\sigma_{j,b}} \frac{\Phi\left(\frac{q_{j,b}-\tilde{m}_{j,b}}{\tilde{s}_{j,b}}\right) - \Phi\left(\frac{q_{j,b-1}-\tilde{m}_{j,b}}{\tilde{s}_{j,b}}\right)}{\Phi\left(\frac{q_{j,b}-\mu_{j,b}}{\sigma_{j,b}}\right) - \Phi\left(\frac{q_{j,b-1}-\mu_{j,b}}{\sigma_{j,b}}\right)} \mathrm{e}^{\frac{-(\mu_{j,b}-\zeta_j)^2}{2(\gamma^2+\sigma_{j,b}^2)}} =: e_{j,b}^k\,. \tag{13}$$

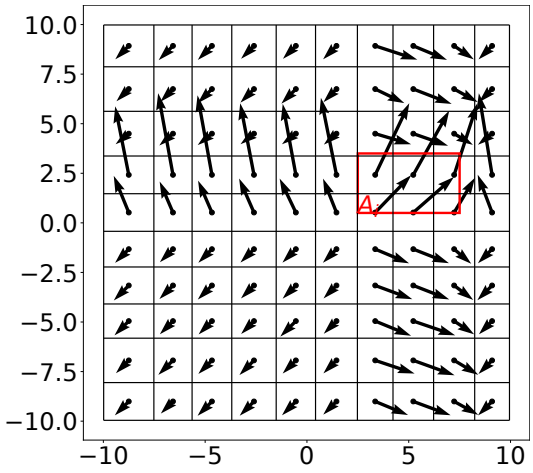

Figure 10: Explanations provided by Tabular LIME for $\mathbb{1}_A$ in dimension 2. In red, the hyperrectangle A. The bins correspond to the discretization of the space with $p = 10$, the training data is a uniform sample on $[-10, 10] \times [-10, 10]$. For each bin, we compute the explanations given by Tabular LIME at the central point. The arrows correspond to the vectors (β_1, β_2) (not to scale): for instance, a large arrow pointing north-east means that both β_1 and β_2 take large positive values at ξ the base-point of the arrow. We see that the interpretable coefficients are small when far away from the hyper-rectangle A. But some artifacts appear one the bins aligned with A along the axes.

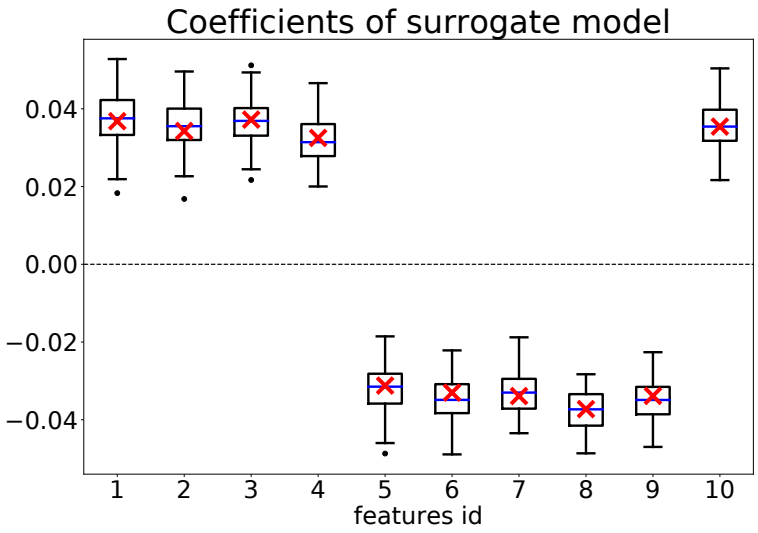

Figure 11: Theory vs practice for f given by a Gaussian kernel with bandwidth parameter $\gamma = 10$. We see that our theoretical predictions (red crosses) match perfectly the values of the interpretable coefficients given by Tabular LIME (100 repetitions, black whisker boxes).

From the previous display we can deduce the value of $e_{j,b}^{f_j}$ for any $1 \leq j \leq d$ and $1 \leq b \leq p$. Namely, $e_{j,b} = e_{j,b}^k$ if $b = b_j^\star$ and $\mathrm{e}^{\frac{-1}{2\nu^2}} e_{j,b}^k$ otherwise. The value of the $c_j^{f_j}$ coefficients follows, which gives us a closed-formula for β_j. Figure 11 demonstrates how our theoretical predictions match practice in dimension 10.

As in the previous section, we can see that a given β_j is relatively larger than the other interpretable coefficients if

$$e_{j,b_j^\star}^k \gg \frac{1}{p-1} \sum_{b \neq b_j^\star} e_{j,b}^k \,.$$

Now, in the typical situation, $\gamma \ll \sigma_{j,b}$ for any $1 \leq j \leq d$ and $1 \leq b \leq p$. Indeed, the bins are usually quite large (containing approximately $1/p$-th of the training data in any given direction), whereas γ should be rather small in order to encompass the small-scale variations of the data. In this case, whenever $\mu_{j,b}$ is far away from ζ_j, the exponential term vanishes, and $e_{j,b}^k \approx 0$. Therefore, we end up in a situation very similar to the indicator function case treated in the previous section, where the only large $e_{j,b}^k$ coefficients are the one for which the associated bin contains ζ_j. This has similar consequences: the explanations are rather small when ξ is far away from ζ. With the same exception: when ξ_j is "aligned" with ζ_j, $e_{j,b}^k$ is large and this yields strange artifacts in the explanation, which we demonstrate in Figure 12. Again, we do not think that this is a desirable behavior. One would prefer to see β_j pointing in the direction of ζ_j, or at the very least small β_j since the function is very flat when far from ζ (at least in the Gaussian kernel case).

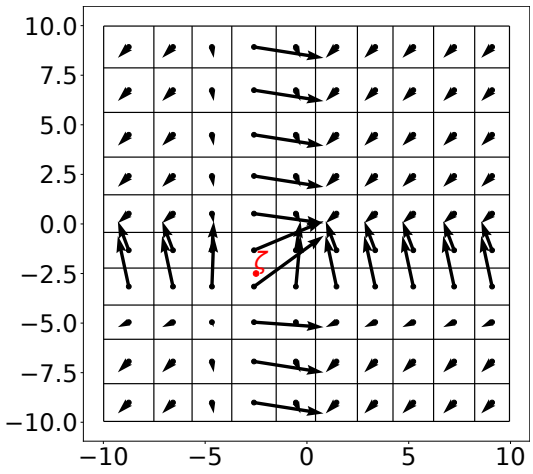

Figure 12: Explanations for a Gaussian kernel in dimension 2 ($\gamma = 1$). As in Figure 10, we considered uniformly distributed training data with $p = 10$ bins along each coordinate. As promised by the theory, the explanations are very small if ξ is far away from ζ, excepted when ξ_j falls into the same bin as ζ_j.

4.3 Model depending on a subset of the coordinates

As a last application of Theorem 1, we turn to the case where f depends only on a (strict) subset of the coordinates. Intuitively, this corresponds to situation where one or more coordinates are unused by the model, and we would like any interpretability method to give a weight 0 to this coordinate—or rather, in the case of Tabular LIME, to the interpretable coefficient along this dimension. We begin with a definition.

Assumption 4.3. Let $s < d$ be a fixed integer. We say that f is s-sparse if there exists $g : \mathbb{R}^s \to \mathbb{R}$ such that

$$\forall x \in \mathbb{R}^d, \quad f(x) = g(x_{j_1}, \ldots, x_{j_s}) \,,$$

where $S := \{j_1, \ldots, j_s\}$ is a subset of $\{1, \ldots, d\}$ of cardinality s.

Our next result shows that, indeed, Tabular LIME discards unused coordinates in its explanation.

Lemma 5 (Ignoring unused coordinates). *Assume that f satisfies Assumption 4.3 and is bounded on $\mathcal{S}$. Let $j \in \overline{S}$, where S is the set of indices relevant for f and $\overline{S} := \{1, \ldots, d\} \setminus S$. Then $\beta_j = 0$.*

We demonstrate Lemma 5 in Figure 13.

While Lemma 5 is true for more general weights (see Appendix I), we provide here a proof for the default weights. Our goal is to demonstrate how Theorem 1 can be used to get interesting statements on the explanations provided by Tabular LIME when minimal structural assumptions are made on f.

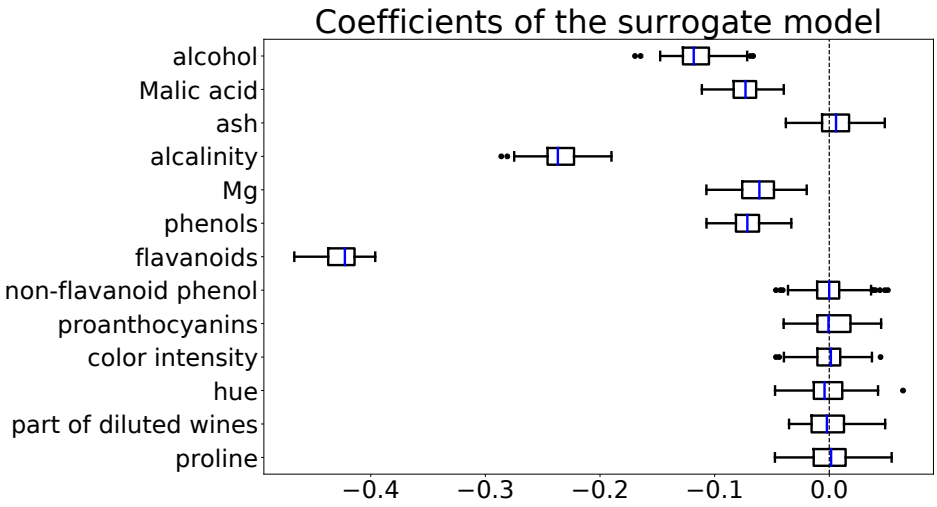

Figure 13: Ignoring unused coordinates. In this experiment, we run Tabular LIME on a kernel ridge regressor (Gaussian kernel with bandwidth set to 5) that does not use the last 6 coordinates. The data is Wine quality. As predicted by Lemma 5, the interpretable coefficients for the last six coordinates are zero up to noise from the sampling: LIME discards these coordinates in the explanation.

Proof. We want to specialize Theorem 1 in the specific case where f does not depend on a subset of coordinates. As we have seen before, the main challenge in using Theorem 1 is to compute the expectations $\mathbb{E}[\pi f(x)]$ and $\mathbb{E}[\pi z_j f(x)]$. The main idea of the proof is to regroup in these expectation computations the parts depending on $j \in S$ and those depending on $j \notin S$. We will see that some cancellations happen afterwards.

Let us turn first to the computation of $\mathbb{E}[\pi f(x)]$. By definition of the weights (Eq. (16)), we have

$$\mathbb{E}[\pi f(x)] = \mathbb{E}\left[\prod_{k=1}^{d} \mathrm{e}^{\frac{-(1-z_k)^2}{2\nu^2}} f(x)\right].$$

Using successively Assumption 4.3 and the independence of the x_js, we can rewrite the previous display as

$$\prod_{k\in\overline{S}} \mathbb{E}\left[\mathrm{e}^{\frac{-(1-z_k)^2}{2\nu^2}}\right] \cdot \mathbb{E}\left[\prod_{k\in S} \mathrm{e}^{\frac{-(1-z_k)^2}{2\nu^2}} g(x_{j_1}, \ldots, x_{j_s})\right].$$

We recognize the definition of the normalization constant c. Setting

$$G := \mathbb{E}\left[\prod_{k\in S} \mathrm{e}^{\frac{-(1-z_k)^2}{2\nu^2}} g(x_{j_1}, \ldots, x_{j_s})\right],$$

we have proved that

$$\mathbb{E}[\pi f(x)] = c^{d-s} \cdot G. \tag{14}$$

Let $j \in \{1 \dots, d\} \setminus S$. The computation of $\mathbb{E}\left[\pi z_j f(x)\right]$ is similar. Indeed, by definition of the weights, we can write

$$\mathbb{E}\left[\pi z_j f(x)\right] = \mathbb{E}\left[\prod_{k=1}^{d} \mathrm{e}^{\frac{-(1-z_k)^2}{2\nu^2}} z_j f(x)\right].$$

Again, since the x_js are independent and f satisfies Assumption 4.2, we rewrite the previous display as

$$\prod_{k \in \overline{S} \setminus \{j\}} \mathbb{E}\left[\mathrm{e}^{\frac{-(1-z_k)^2}{2\nu^2}}\right] \cdot \mathbb{E}\left[\mathrm{e}^{\frac{-(1-z_j)^2}{2\nu^2}} z_j\right] \cdot \mathbb{E}\left[\prod_{k \in S} \mathrm{e}^{\frac{-(1-z_k)^2}{2\nu^2}} g(x_{j_1}, \dots, x_{j_s})\right].$$

We have proved that

$$\mathbb{E}\left[\pi z_j f(x)\right] = \frac{c^{d-s-1} G}{p}. \tag{15}$$

Finally, according to Theorem 1,

$$\beta_j = C^{-1} \frac{pc}{pc-1} \left(- \mathbb{E}\left[\pi f(x)\right] + pc\, \mathbb{E}\left[\pi z_j f(x)\right] \right).$$

Plugging Eqs. (14) and (15) in the previous display, we obtain that

$$\beta_j = C^{-1} \frac{pc}{pc-1} \left(-c^{d-s} \cdot G + pc \frac{c^{d-s-1} G}{p} \right) = 0\,.$$

□

5. Conclusion: strengths and weaknesses of Tabular LIME

In this paper, we provided a thorough analysis of Tabular LIME. In particular, we show that, in the large sample size, the interpretable coefficients provided by Tabular LIME can be obtained in an explicit way as a function of the algorithm parameters and some expectation computations related to the black-box model. Our experiments show that our theoretical analysis yields predictions that are very close from empirical results for the default implementation. This allowed us to provide very precise insights on the inner working of Tabular LIME, revealing some desirable behavior (linear model are explained linearly, unused coordinates are provably ignored, and kernel functions yield flat explanations far away from the bump), and some not so desirable (artifacts appear when explaining kernel functions).

We believe that the present work paves the way for a better theoretical understanding of Tabular LIME in numerous simple cases. Using the machinery presented here, one can check how a particular model interacts with Tabular LIME, at the price of reasonable computations. It is then possible to check whether some basic sanity checks are satisfied, helping us to decide whether to use Tabular LIME in this case. We also hope that the insights presented here can help us to design better interpretability methods by fixing the flaws of Tabular LIME. For instance, our analysis is valid for a large class of weights: for

a given class of models, it could be the case that choosing non-default weights is more adequate.

Finally, note that we focused on tabular data since LIME is easier to analyze than the text and image version but we would like to tackle the text and image version. Indeed, while the sampling of the perturbed examples is similar in all three versions, they are some key differences which make the analysis even more challenging if the data is not tabular. Namely, in the text version, the TF-IDF transform (Jones, 1972) is used as data transformation between the text and the model. Its non-linear nature makes it hard to analyze. In the image version, the first step of the algorithm is to create superpixels of the image to explain. It is a complicated process which we do not know how to formalize properly.

Acknowledgments

The authors want to thank Romaric Gaudel, Michael Lohaus, and Martin Pawelcyk for constructive discussions.

References

A. Adadi and M. Berrada. Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). *IEEE Access*, 6:52138–52160, 2018.

D. Alvarez-Melis and T. S. Jaakkola. On the robustness of interpretability methods. *arXiv preprint arXiv:1806.08049*, 2018.

G. Biau and E. Scornet. A random forest guided tour. *Test*, 25(2):197–227, 2016.

S. Boucheron, G. Lugosi, and P. Massart. *Concentration inequalities: A nonasymptotic theory of independence*. Oxford university press, 2013.

L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. Classification and regression trees, 1984.

T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, pages 1721–1730, 2015.

P. Cortez, A. Cerdeira, F. Almeida, T. Matos, and J. Reis. Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47(4):547–553, 1998.

D. Garreau and U. von Luxburg. Explaining the explainer: A first theoretical analysis of lime. In *Proceedings of the 33rd International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 1287–1296. PMLR, 2020.

R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5): 1–42, 2018.

D. Harrison Jr. and D. L. Rubinfeld. Hedonic housing prices and the demand for clean air. *Journal of Environmental Economics and Management*, pages 81–102, 1978.

T. Hastie, R. Tibshirani, and R. Friedman. *The Elements of Statistical Learning*. Springer, second edition, 2001.

T. J. Hastie and R. J. Tibshirani. *Generalized additive models*, volume 43. CRC press, 1990.

K. S. Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 1972.

M. S. Kovalev, L. V. Utkin, and E. M. Kasimov. Survlime: A method for explaining machine learning survival models. *arXiv preprint arXiv:2003.08371*, 2020.

S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pages 4765–4774, 2017.

S. Mishra, B. L. Sturm, and S. Dixon. Local interpretable model-agnostic explanations for music content analysis. In *ISMIR*, pages 537–543, 2017.

M. T. Ribeiro, S. Singh, and C. Guestrin. Why should I trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, pages 1135–1144. ACM, 2016.

M. T. Ribeiro, S. Singh, and C. Guestrin. Anchors: High-precision model-agnostic explanations. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

B. Schoelkopf and A. J. Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. the MIT Press, 2001.

L. S. Shapley. A value for n-person games. *Contributions to the Theory of Games*, 2(28): 307–317, 1953.

C. J. Stone. Additive regression and other nonparametric models. *The Annals of Statistics*, pages 689–705, 1985.

J. A. Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of computational mathematics*, 12(4):389–434, 2012.

R. Turner. A model explanation system. In *IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, 2016.

L. V. Utkin, M. S. Kovalev, and E. M. Kasimov. Survlime-inf: A simplified modification of survlime for explanation of machine learning survival models. *arXiv preprint arXiv:2005.02387*, 2020.

R. Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

S. Wachter, B. Mittelstadt, and L. Floridi. Why a right to explanation of automated decision-making does not exist in the general data protection regulation. *International Data Privacy Law*, 7(2):76–99, 2017.

Appendix

In this Appendix we collect all proofs and additional results. Appendix A generalizes the notation to more general weights. In Appendix B, C, and D we study successively Σ, Γ, and β. The proof of Lemma 2 is presented in Appendix E. We turn to the concentration of $\hat{\Sigma}_n$ and $\hat{\Gamma}_n$ in Appendix F and G. Our main result is proved in Appendix H. An extension of Lemma 5 for general weights is presented in Appendix I. Finally, technical results are collected in Appendix J.

Appendix A. General weights

As discussed throughout the main paper, our analysis holds for more general weights than the default weighting scheme. Indeed, it will become clear in our analysis that the proof scheme works provided that the weights π_i have some multiplicative structure. More precisely, from now on we assume that

$$\pi_i := \exp\left(\frac{-1}{2\nu^2}\sum_{j=1}^{d}(\tau_j(\xi_j) - \tau_j(x_{ij}))^2\right), \tag{16}$$

where $\nu > 0$ is, as before, the bandwidth parameter, and $\tau_j : \mathbb{R} \to \mathbb{R}$ are arbitrary fixed functions that can depend on the input of the algorithm. We will refer to these weights as *general weights* in the following. We make the following assumption on the τ_js:

Assumption A.1. For any $1 \leq j \leq d$ and any $x, y \in [0,1]$, we have

$$|\tau_j(x) - \tau_j(y)| \leq 1\,.$$

This assumption is mainly needed to control the spectrum of Σ.

General weights generalize two important examples, which we describe below.

Example A.1 (Weights in the default implementation). In the default implementation of LIME, for a given example x_i, we have defined $\pi_i = \exp\left(-\left\|\mathbb{1} - z_i\right\|^2/(2\nu^2)\right)$ (Eq (2)). This definition of the weights corresponds to taking

$$\tau_j(x) = \mathbb{1}_{x \in \left[q_{j,b_j^\star - 1}, q_{j,b_j^\star}\right]}\,.$$

By definition of the τ_js in this case, Assumption A.1 is satisfied.

Example A.2 (Smooth weights). In Garreau and von Luxburg (2020), the weights were defined as $\pi_i = \exp\left(-\left\|\xi - x_i\right\|^2/(2\nu^2)\right)$. This definition of the weights corresponds to taking $\tau_j(x) = x$ for any $1 \leq j \leq d$. If the data is bounded, then the boundedness of the τ_js is satisfied in this case.

Appendix B. The study of Σ

In this section, we study the matrix Σ for arbitrary weights satisfying Eq. (16). We begin by generalizing the notation $e^{\psi}_{j,b}$ introduced at the beginning of Section 4 to general weights. For any $\psi : \mathbb{R} \to \mathbb{R}$, we set

$$e^{\psi}_{j,b} := \mathbb{E}\left[\psi(x_{ij}) \exp\left(\frac{-1}{2\nu^2}(\tau_j(\xi_j) - \tau_j(x_{ij}))^2\right) \middle| b_{ij} = b\right] . \tag{17}$$

As before, when $\psi = 1$, we just write $e_{j,b}$ instead of $e^1_{j,b}$, and when $\psi = \mathrm{id}$, we write $e^x_{j,b}$. We also extend the definition of the normalization constants

$$c^{\psi}_j := \frac{1}{p}\sum_{b=1}^{p} e^{\psi}_{j,b} ,$$

and $c_j := c^1_j$. Finally, we set $C := \prod_{j=1}^{d} c_j$.

Remark B.1. Whenever ψ is regular enough, the behavior of these coefficients in the small and large bandwidth is quite straightforward. Namely:

- if $\nu \to 0$, then $e^{\psi}_{j,b} \to 0$. As a consequence, $c^{\psi}_j \to 0$ as well;
- if $\nu \to +\infty$, then $e^{\psi}_{j,b} \to \mathbb{E}\left[\psi(x_{ij}) | b_{ij} = b\right]$.

In Section 4, we computed these coefficients for the default weights. Let us redo this exercise for the weighting scheme of Garreau and von Luxburg (2020).

Example B.1 (Basic computations, smooth weights). Let us compute the $e_{j,b}$ when $\tau_j = \mathrm{id}$. We write

$$\begin{aligned}
e_{j,b} &= \mathbb{E}\left[\exp\left(\frac{-(x_{ij} - \xi_j)^2}{2\nu^2}\right) \middle| b_{ij} = b\right] && \text{(Eq. (17))} \\
&= \frac{1}{\sigma_{jb}\sqrt{2\pi}} \cdot \frac{1}{\Phi(r_{j,b}) - \Phi(\ell_{j,b})} \int_{q_{j,b-1}}^{q_{j,b}} \exp\left(\frac{-(x-\mu_{j,b})^2}{2\sigma^2_{j,b}} + \frac{-(x-\xi_j)^2}{2\nu^2}\right) \mathrm{d}t && \text{(Eq. (1))} \\
&= \frac{1}{\Phi(r_{j,b}) - \Phi(\ell_{j,b})} \cdot \frac{\nu\, \mathrm{e}^{\frac{-(\xi_j - \mu_{j,b})^2}{2(\nu^2 + \sigma^2_{j,b})}}}{\sqrt{\nu^2 + \sigma^2_{j,b}}} \left[\frac{1}{2}\mathrm{erf}\left(\frac{\nu^2(x-\mu_{j,b}) + \sigma^2_{j,b}(x - \xi_j)}{\nu\sigma_{j,b}\sqrt{2(\nu^2 + \sigma^2_{j,b})}}\right)\right]_{q_{j,b-1}}^{q_{j,b}} ,
\end{aligned}$$

where we used Lemma 11.1 in Garreau and von Luxburg (2020) in the last display. Let us set

$$m_{j,b} := \frac{\nu^2\mu_{j,b} + \sigma^2_{j,b}\xi_j}{\nu^2 + \sigma^2_{j,b}} \quad \text{and} \quad s^2_{j,b} := \frac{\nu^2\sigma^2_{j,b}}{\nu^2 + \sigma^2_{j,b}} .$$

Then it is straightforward to show that

$$\frac{\nu^2(x-\mu_{j,b}) + \sigma^2_{j,b}(x-\xi_j)}{\nu\sigma_{j,b}\sqrt{2(\nu^2 + \sigma^2_{j,b})}} = \frac{x - m_{j,b}}{\sqrt{2}s_{j,b}} ,$$

and the expression of $e_{j,b}$ simplifies slightly:

$$e_{j,b} = \frac{1}{\Phi(r_{j,b}) - \Phi(\ell_{j,b})} \cdot \frac{\nu \, \mathrm{e}^{\frac{-(\xi_j - \mu_{j,b})^2}{2(\nu^2 + \sigma_{j,b}^2)}}}{\sqrt{\nu^2 + \sigma_{j,b}^2}} \left[\frac{1}{2} \mathrm{erf}\left(\frac{x - m_{j,b}}{\sqrt{2} s_{j,b}} \right) \right]_{q_{j,b-1}}^{q_{j,b}} .$$

Now recall that Garreau and von Luxburg (2020) chose to consider $\mu_{j,b} = \mu_j$ and $\sigma_{j,b} = \sigma$ constant. As a consequence, $m_{j,b}$ does not depend on b anymore, and $s_{j,b}$ is a constant equal to $s := \nu\sigma/\sqrt{\nu^2 + \sigma^2}$. Moreover, the $q_{j,b}$ are, in this case, the normalized Gaussian quantiles, and therefore

$$\Phi(r_{j,b}) - \Phi(\ell_{j,b}) = \frac{1}{p} .$$

We deduce that

$$e_{j,b} = \frac{p\nu \, \mathrm{e}^{\frac{-(\xi_j - \mu_j)^2}{2(\nu^2 + \sigma^2)}}}{\sqrt{\nu^2 + \sigma^2}} \left[\frac{1}{2} \mathrm{erf}\left(\frac{x - m_j}{\sqrt{2} s} \right) \right]_{q_{j,b-1}}^{q_{j,b}} ,$$

and

$$c_j = \frac{\nu}{\sqrt{\nu^2 + \sigma^2}} \exp\left(\frac{-(\xi_j - \mu_j)^2}{2(\nu^2 + \sigma^2)} \right) ,$$

which yields

$$C = \left(\frac{\nu^2}{\nu^2 + \sigma^2} \right)^{d/2} \exp\left(\frac{-\left\| \xi - \mu \right\|^2}{2(\nu^2 + \sigma^2)} \right) . \tag{18}$$

We see that Eq. (18) is indeed Eq. (7.2) in Garreau and von Luxburg (2020).

B.1 Computation of Σ

We now have the necessary notation to turn to the computation of Σ.

Lemma 6 (Computation of Σ, arbitrary weights). *Recall that $\hat{\Sigma}_n = \frac{1}{n} Z^\top W Z$ and $\Sigma = \mathbb{E}[\hat{\Sigma}_n]$, where Z was defined in Section 2.2. Then it holds that*

$$\Sigma = C \begin{pmatrix} 1 & \frac{e_{1,b_1^\star}}{pc_1} & \cdots & \frac{e_{d,b_d^\star}}{pc_d} \\ \frac{e_{1,b_1^\star}}{pc_1} & \frac{e_{1,b_1^\star}}{pc_1} & & \frac{e_{j,b_j^\star} e_{k,b_k^\star}}{pc_j pc_k} \\ \vdots & & \ddots & \\ \frac{e_{d,b_d^\star}}{pc_d} & \frac{e_{j,b_j^\star} e_{k,b_k^\star}}{pc_j pc_k} & & \frac{e_{d,b_d^\star}}{pc_d} \end{pmatrix}$$

Proof. We introduce a phantom coordinate in Z since the surrogate linear model uses an offset. A straightforward computation shows that

$$\hat{\Sigma} = \frac{1}{n} \begin{pmatrix} \sum_{i=1}^n \pi_i & \sum_{i=1}^n \pi_i z_{i1} & \cdots & \sum_{i=1}^n \pi_i z_{id} \\ \sum_{i=1}^n \pi_i z_{i1} & \sum_{i=1}^n \pi_i z_{i1} & & \sum_{i=1}^n \pi_i z_{ij} z_{ik} \\ \vdots & & \ddots & \\ \sum_{i=1}^n \pi_i z_{id} & \sum_{i=1}^n \pi_i z_{ij} z_{ik} & & \sum_{i=1}^n \pi_i z_{i1} \end{pmatrix} . \tag{19}$$

Since the x_is are i.i.d., the result follows from the computation of $\mathbb{E}[\pi]$, $\mathbb{E}[\pi z_j]$, and $\mathbb{E}[\pi z_j z_k]$. This is achieved in Lemma 28. □

As examples, we can explicit the computation of Σ for default weights and the weighting scheme of Garreau and von Luxburg (2020).

Example B.2 (Default implementation, computation of Σ). As we have seen in Section 4, in this case

$$e_{j,b} = \begin{cases} 1 \text{ if } b = b_j^\star \\ \mathrm{e}^{\frac{-1}{2\nu^2}} \text{ otherwise,} \end{cases}$$

and $c_j = c$ is a constant. Therefore, according to Lemma 6, the expression of Σ simplifies greatly into

$$\Sigma = c^d \begin{pmatrix} 1 & \frac{1}{pc} & \cdots & \frac{1}{pc} \\ \frac{1}{pc} & \frac{1}{pc} & & \frac{1}{p^2c^2} \\ \vdots & & \ddots & \\ \frac{1}{pc} & \frac{1}{p^2c^2} & & \frac{1}{pc} \end{pmatrix}.$$

Example B.3 (Computation of Σ, smooth weights). According to Example B.1, we have that

$$\frac{e_{j,b_j^\star}}{pc_j} = \left[\frac{1}{2}\mathrm{erf}\left(\frac{x - m_j}{\sqrt{2}s}\right)\right]_{q_{j,b_j^\star - 1}}^{q_{j,b_j^\star}}.$$

We recover the α_j coefficients (Eq. (7.3) in Garreau and von Luxburg (2020)) and as a direct consequence, Lemma 8.1:

$$\Sigma := C \begin{pmatrix} 1 & \alpha_1 & \cdots & \alpha_d \\ \alpha_1 & \alpha_1 & & \alpha_i\alpha_j \\ \vdots & & \ddots & \\ \alpha_d & \alpha_i\alpha_j & & \alpha_d \end{pmatrix}.$$

B.2 Computation of Σ^{-1}

The structure of Σ allows to invert it in closed-form, even in the case of general weights. This is the extent of the next lemma.

Lemma 7 (Computation of Σ^{-1}, arbitrary weights). *Let Σ be defined as before. Then Σ is invertible and*

$$\Sigma^{-1} = C^{-1} \begin{pmatrix} 1 + \sum_{j=1}^d \frac{e_{j,b_j^\star}}{pc_j - e_{j,b_j^\star}} & \frac{-pc_1}{pc_1 - e_{1,b_1^\star}} & \cdots & \frac{-pc_d}{pc_d - e_{d,b_d^\star}} \\ \frac{-pc_1}{pc_1 - e_{1,b_1^\star}} & \frac{p^2c_1^2}{e_{1,b_1^\star}(pc_1 - e_{1,b_1^\star})} & & 0 \\ \vdots & & \ddots & \\ \frac{-pc_d}{pc_d - e_{d,b_d^\star}} & 0 & & \frac{p^2c_d^2}{e_{d,b_d^\star}(pc_d - e_{d,b_d^\star})} \end{pmatrix}.$$

Proof. Set $\alpha_j := e_{j,b_j^\star}/(pc_j)$, and define $\alpha \in \mathbb{R}^d$ the vector of the α_js. Set $A := 1$, $B := \alpha^\top$, $C := \alpha$, and

$$D := \begin{pmatrix} \alpha_1 & & \alpha_j\alpha_k \\ & \ddots & \\ \alpha_j\alpha_k & & \alpha_d \end{pmatrix} .$$

Then Σ is a block matrix that can be written $\Sigma = C \begin{bmatrix} A & B \\ C & D \end{bmatrix}$. We notice that

$$D - CA^{-1}B = \operatorname{diag}\left(\alpha_1(1-\alpha_1), \dots, \alpha_d(1-\alpha_d)\right) .$$

Since the $e_{j,b}$ are positive for any j, b, the α_js belong to $(0,1)$ and $D - CA^{-1}B$ is an invertible matrix. According to the block matrix inversion formula,

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{pmatrix} A^{-1} + A^{-1}B(D-CA^{-1}B)^{-1}CA^{-1} & -A^{-1}B(D-CA^{-1}B)^{-1} \\ -(D-CA^{-1}B)^{-1}CA^{-1} & (D-CA^{-1}B)^{-1} \end{pmatrix}$$

Thus

$$\Sigma^{-1} = C^{-1} \begin{pmatrix} 1 + \sum_{j=1}^d \frac{\alpha_j}{1-\alpha_j} & \frac{-1}{1-\alpha_1} & \cdots & \frac{-1}{1-\alpha_d} \\ \frac{-1}{1-\alpha_1} & \frac{1}{\alpha_1(1-\alpha_1)} & & 0 \\ \vdots & & \ddots & \\ \frac{-1}{1-\alpha_d} & 0 & & \frac{1}{\alpha_d(1-\alpha_d)} \end{pmatrix} . \tag{20}$$

The result follows from the definition of the α_js. □

We can be explicit about the computation of Σ^{-1} in the case of default and smooth weights.

Example B.4 (Computation of Σ^{-1}, default implementation). Using our basic computations in this case in conjunction with Lemma 7, we obtain

$$\Sigma^{-1} = \frac{c^{-d}}{pc-1} \begin{pmatrix} pc+d-1 & -pc & \cdots & -pc \\ -pc & p^2c^2 & & 0 \\ \vdots & & \ddots & \\ -pc & 0 & & p^2c^2 \end{pmatrix} .$$

Example B.5 (Computation of Σ^{-1}, smooth weights). From the definition of α_j in Example B.1, we see that Eq. (20) is in fact Lemma 8.2 in Garreau and von Luxburg (2020).

B.3 Control of $\left\|\Sigma^{-1}\right\|_{\text{op}}$

Our analysis requires a control of $\left\|\Sigma^{-1}\right\|_{\text{op}}$ when we want to concentrate $\hat{\beta}_n$ (see Appendix H). We show how to achieve this control when the functions τ_j are bounded.

Lemma 8 (Upper bound on $\left\|\Sigma^{-1}\right\|_{\mathrm{op}}$, general weights). *Let Σ be as before, and suppose that τ_j satisfies Assumption A.1. Then*

$$\left\|\Sigma^{-1}\right\|_{\mathrm{op}} \leq 2\sqrt{2}C^{-1}dp^2\,\mathrm{e}^{\frac{2}{\nu^2}}\ .$$

Proof. According to Lemma 31, we can find an upper bound for $\left\|\Sigma^{-1}\right\|_{\mathrm{op}}$ just by computing $\left\|\Sigma^{-1}\right\|_{\mathrm{F}}$. Lemma 7 gives us

$$\left\|C\Sigma^{-1}\right\|_{\mathrm{F}}^2 = \left(1+\sum_{j=1}^{d}\frac{e_{j,b_j^\star}}{pc_j-e_{j,b_j^\star}}\right)^2 + 2\sum_{j=1}^{d}\frac{p^2c_j^2}{(pc_j-e_{j,b_j^\star})^2} + \sum_{j=1}^{d}\frac{p^4c_j^4}{e_{j,b_j^\star}(pc_j-e_{j,b_j^\star})^2}\,. \quad (21)$$

We first notice that all the terms involved in Eq. (21)are positive. Moreover, we see that $e_{j,b} \leq 1$ and $pc_j \leq p$ for any j, b. Under Assumption A.1, $|\tau_j(\xi_j) - \tau_j(x_{ij})| \leq 1$ almost surely. We deduce that

$$e_{j,b} = \mathbb{E}\left[\exp\left(\frac{-1}{2\nu^2}(\tau_j(\xi_j)-\tau_j(x_{ij}))^2\right)\middle| b_{ij}=b\right] \geq \mathrm{e}^{\frac{-1}{2\nu^2}}\ .$$

As a consequence, $pc_j - e_{j,b_j^\star} \geq (p-1)\,\mathrm{e}^{\frac{-1}{2\nu^2}}$. Plugging these bounds in Eq. (21) and using $(x+y)^2 \leq 2(x^2+y^2)$, we obtain

$$\left\|C\Sigma^{-1}\right\|_{\mathrm{F}}^2 \leq 2\left(1+\frac{d^2\,\mathrm{e}^{\frac{1}{\nu^2}}}{(p-1)^2}\right) + \frac{2dp^2\,\mathrm{e}^{\frac{1}{\nu^2}}}{(p-1)^2} + \frac{dp^4\,\mathrm{e}^{\frac{4}{\nu^2}}}{(p-1)^4}\,.$$

Finally, we conclude using $p \geq 2$ and $d \geq 1$. □

Of course, a more precise knowledge of the weights can give a better bound for $\left\|\Sigma^{-1}\right\|_{\mathrm{op}}$. For instance, it is possible to show that

$$\left\|\Sigma^{-1}\right\|_{\mathrm{op}} \lesssim c^{-d}\,\mathrm{e}^{\frac{1}{2\nu^2}}\,(d+p)$$

in that case, by studying closely the spectrum of Σ. Since the difference with Lemma 8 is not that impressive, we keep the bound given for general weights.

Appendix C. The study of Γ

In this section, we turn to the study of Γ, the second brick in our analysis. In the previous section, there was no dependency in f, this is not the case anymore. We will assume that f is bounded on $\mathcal{S}$ from now on, as in the assumptions of our main result (Theorem 24). In particular, this assumption guarantees that all the expectations involving f are well defined (since π and z are also bounded almost surely).

C.1 Computation of Γ

We begin by computing Γ. A straightforward computation shows that

$$\hat{\Gamma}_j = \begin{cases} \frac{1}{n}\sum_{i=1}^n \pi_i f(x_i) \text{ if } j = 0 \\ \frac{1}{n}\sum_{i=1}^n \pi_i z_{ij} f(x_i) \text{ otherwise.} \end{cases}$$

Since the x_i are i.i.d., a straightforward consequence of the previous display is

$$\Gamma_j = \begin{cases} \mathbb{E}\left[\pi f(x)\right] \text{ if } j = 0 \\ \mathbb{E}\left[\pi z_j f(x)\right] \text{ otherwise.} \end{cases} \tag{22}$$

Example C.1 (Constant model, general weights). In order to get familiar with Eq. (22), let us focus on a constant model. In this case, we just need to compute $\mathbb{E}\left[\pi\right]$ and $\mathbb{E}\left[\pi z_j\right]$. According to Lemma 28, we have

$$\mathbb{E}\left[\pi\right] = C \quad \text{and} \quad \mathbb{E}\left[\pi z_j\right] = C\frac{e_{j,b_j^\star}}{pc_j}\,.$$

We have just showed that, if $f = f_0$ a constant, then

$$C^{-1}\Gamma_0 = f_0 \quad \text{and} \quad C^{-1}\Gamma_j = \frac{e_{j,b_j^\star}}{pc_j} f_0 \quad \forall j \geq 1\,. \tag{23}$$

C.2 Additive f

We can specialize the computation of Γ when f is additive.

Lemma 9 (Computation of Γ, additive f). *Assume that f satisfies Assumption 4.1 and that f is bounded on $\mathcal{S}$. Then Γ is such that*

$$\Gamma_0 = \sum_{k=1}^d \frac{1}{pc_k} \sum_{b=1}^p e_{k,b}^{f_k}\,,$$

and, for any $1 \leq j \leq d$,

$$\Gamma_j = \sum_{k=1}^d \frac{e_{j,b_j^\star}}{pc_j} \cdot \frac{1}{pc_k} \sum_{b=1}^p e_{k,b}^{f_k} + \frac{1}{pc_j}\left[e_{j,b_j^\star}^{f_j} - \frac{e_{j,b_j^\star}}{pc_j}\sum_{b=1}^p e_{j,b}^{f_j}\right].$$

Proof. By linearity and since f is additive, we can restrict ourselves to the case $f(x) = \psi(x_k)$. We first look into $j = 0$ in Eq. (22). Then

$$\mathbb{E}\left[\pi_i f(x_i)\right] = \mathbb{E}\left[\pi_i \psi(x_{ik})\right] = \frac{C}{pc_k}\sum_{b=1}^p e_{k,b}^{\psi}$$

according to Lemma 27. Setting $\psi = f_k$ in the previous display and summing for $k \in \{1, \ldots, d\}$, we obtain the first part of the result.

Now we turn to the case $j \geq 1$. Again, by linearity and since f is additive, we can restrict ourselves to the case $f(x) = \psi(x_k)$. There are two possibilities in this case: either $j = k$, and then

$$\mathbb{E}\left[\pi_i z_{ij} f(x_i)\right] = \mathbb{E}\left[\pi_i z_{ij} \psi(x_{ij})\right] = C \frac{e^{\psi}_{j,b^\star_j}}{pc_j},$$

according to Lemma 27; or $j \neq k$, and then

$$\mathbb{E}\left[\pi_i z_{ij} f(x_i)\right] = \mathbb{E}\left[\pi_i z_{ij} \psi(x_{ik})\right] = C \frac{e_{j,b^\star_j}}{pc_j} \frac{1}{pc_k} \sum_{b=1}^{p} e^{\psi}_{k,b},$$

according to Lemma 27. Setting $\psi = f_k$ in the last displays and summing over $k \in \{1, \ldots, d\}$, we obtain

$$C^{-1}\, \mathbb{E}\left[\pi_i z_{ij} f(x_i)\right] = \sum_{\substack{k=1 \\ k \neq j}}^{d} \frac{e_{j,b^\star_j}}{pc_j} \frac{1}{pc_k} \sum_{b=1}^{p} e^{\psi}_{k,b} + \frac{e^{f_j}_{j,b^\star_j}}{pc_j}.$$

Rearranging the terms in the sum yields the final result. □

Let us specialize Lemma 9 for default weights.

Example C.2 (Computation of Γ, additive f, default weights). We can use Lemma 9 to compute Γ for an additive f when the weights are given by the default implementation. Indeed, recall that $\mathbb{E}\left[x_{ij} | b_{ij} = b\right] = \tilde{\mu}_{j,b}$. Since the weights are constant on each box, we can compute easily $e^x_{j,b}$ as a function of $\tilde{\mu}_{j,b}$:

$$e^x_{j,b} = \begin{cases} \tilde{\mu}_{j,b^\star_j} \text{ if } b = b^\star_j \\ \tilde{\mu}_{j,b}\, \mathrm{e}^{\frac{-1}{2\nu^2}} \text{ otherwise.} \end{cases}$$

Also recall that the c_j are constant equal to $c = \frac{1}{p} + (1 - \frac{1}{p})\, \mathrm{e}^{\frac{-1}{2\nu^2}}$. We deduce that

$$(c^{-d}\Gamma)_0 = \sum_{k=1}^{d} \frac{1}{pc} \sum_{b=1}^{p} e^{f_k}_{k,b},$$

and, for any $1 \leq j \leq d$,

$$(c^{-d}\Gamma)_j = \sum_{k=1}^{d} \frac{e_{j,b^\star_j}}{pc} \cdot \frac{1}{pc} \sum_{b=1}^{p} e^{f_k}_{k,b} + \frac{1}{pc} \left(e^{f_j}_{j,b^\star_j} - \frac{1}{pc} \sum_{b=1}^{p} e^{f_j}_{j,b} \right).$$

We can specialize Lemma 9 even further if f is linear.

Lemma 10 (Computation of Γ, linear case, general weights). *Assume that $f(x) = f_0 + f_1 x_1 + \cdots + f_d x_d$ for any $x \in \mathbb{R}^d$. Then Γ is such that*

$$\Gamma_0 = f(\gamma),$$

and, for any $1 \leq j \leq d$,

$$\frac{e_{j,b_j^\star}}{pc_j} f(\gamma) + \frac{f_j}{pc_j} \left[e^x_{j,b_j^\star} - \gamma_j \cdot e_{j,b_j^\star} \right] ,$$

where we defined, for any $1 \leq j \leq d$,

$$\gamma_j := \frac{1}{pc_j} \sum_{b=1}^{p} e^x_{j,b} \, .$$

Proof. Again, we use the fact that Γ is linear as a function of f. We have already looked into the constant case, and one can check that Eq. (23) coincides with Lemma 10 when $f_j = 0$ for all $j \geq 1$. We then apply Lemma 9 with $f_j(x) = f_j \cdot x$ for any $j \geq 1$. We notice that, in this case, for any $j \in \{1, \dots, d\}$ and $b \in \{1, \dots, p\}$,

$$e^{f_j}_{j,b} = f_j \cdot e_{j,b} \, .$$

Substituting the last display yields the result. □

Let us specialize Lemma 10 for default weights and smooth weights.

Example C.3 (Computation of Γ, linear f, default weights). We can further specialize Example C.2. The only remaining computation is

$$\gamma_j = \frac{1}{pc_j} \sum_{b=1}^{p} e^x_{j,b} = \frac{\tilde{\mu}_{j,b_j^\star} + \sum_{b \neq b_j^\star} \mathrm{e}^{\frac{-1}{2\nu^2}} \, \tilde{\mu}_{j,b}}{1 + (p-1) \, \mathrm{e}^{\frac{-1}{2\nu^2}}} =: \tilde{\tilde{\mu}}_j \, .$$

Note that $\tilde{\tilde{\mu}}_j$ is a barycenter of the $\tilde{\mu}_j$ with weight 1 for the box $b_j^\star$ and $\mathrm{e}^{\frac{-1}{2\nu^2}}$ for the others. Finally, recall that $C = c^d$ and $e_{j,b_j^\star} = 1$. We have obtained that

$$c^{-d} \Gamma_j = \begin{cases} f(\tilde{\tilde{\mu}}) \text{ if } j = 0 \\ \frac{1}{pc} f(\tilde{\tilde{\mu}}) + \frac{f_j}{pc} (\tilde{\mu}_{j,b_j^\star} - \tilde{\tilde{\mu}}_j) \text{ otherwise.} \end{cases}$$

Example C.4 (Computation of Γ, linear f, smooth weights). We first compute

$$\begin{aligned} e^x_{j,b_j^\star} &= \mathbb{E}\left[x_{ij} \exp\left(\frac{-(x_{ij} - \xi_j)^2}{2\nu^2} \right) \middle| b_{ij} = b_j^\star \right] \\ &= \frac{p}{\sigma\sqrt{2\pi}} \int_{q_{j,b_j^\star - 1}}^{q_{j,b_j^\star}} x \cdot \exp\left(\frac{-(x - \xi_j)^2}{2\nu^2} + \frac{-(x - \mu_j)^2}{2\sigma^2} \right) \mathrm{d}x \qquad \text{(Eq. (1))} \\ &= (\alpha_j m_j - \theta_j) \cdot \frac{p\nu}{\sqrt{\nu^2 + \sigma^2}} \, \mathrm{e}^{\frac{-(\xi_j - \mu_j)^2}{2(\nu^2 + \sigma^2)}} \, , \end{aligned}$$

(Lemma 11.2 in Garreau and von Luxburg (2020))

where we set

$$\theta_j := \left[\frac{1}{s\sqrt{2\pi}} \exp\left(\frac{-(x - m_j)^2}{2s^2} \right) \right]_{q_{j,b_j^\star - 1}}^{q_{j,b_j^\star}} \, .$$

We deduce that

$$\frac{e^x_{j,b^\star_j}}{pc_j} = \alpha_j m_j - \theta_j \,.$$

Moreover, by the law of total expectation and Lemma 11.2 in Garreau and von Luxburg (2020),

$$\frac{1}{p}\sum_{b=1}^{p} e^x_{j,b} = m_j \cdot \frac{\nu}{\sqrt{\nu^2+\sigma^2}} \exp\left(\frac{-(\xi_j-\mu_j)^2}{2(\nu^2+\sigma^2)}\right) .$$

Finally, since $e_{j,b^\star_j}/(pc_j) = \alpha_j$ and $\gamma_j = m_j$, we find that $\Gamma_0 = f(m)$ and

$$\Gamma_j = \alpha_j f(m) + f_j(\alpha_j m_j - \theta_j - \alpha_j m_j) = \alpha_j f(m) - f_j\theta_j \,.$$

We recover Lemma 9.1 in Garreau and von Luxburg (2020).

Appendix D. The study of β

After focusing on Σ and Γ, we can now turn our attention to $\beta = \Sigma^{-1}\Gamma$. This section consists mostly in easy consequences of the computations achieved in Appendix B and C.

D.1 Computation of β

We begin by computing β in the general case in closed-form.

Lemma 11 (Computation of β, general f, general weights). *Assume that f is bounded on $\mathcal{S}$. Then it holds that*

$$\beta_0 = C^{-1}\left(1 + \sum_{j=1}^{d} \frac{e_{j,b^\star_j}}{pc_j - e_{j,b^\star_j}}\right)\mathbb{E}\left[\pi f(x)\right] - C^{-1}\sum_{j=1}^{d}\frac{pc_j}{pc_j - e_{j,b^\star_j}}\mathbb{E}\left[\pi z_j f(x)\right] ,$$

and, for any $1 \leq j \leq d$,

$$\beta_j = C^{-1}\left(\frac{-pc_j}{pc_j - e_{j,b^\star_j}}\mathbb{E}\left[\pi f(x)\right] + \frac{p^2c_j^2}{e_{j,b^\star_j}(pc_j - e_{j,b^\star_j})}\mathbb{E}\left[\pi z_j f(x)\right]\right) .$$

Proof. Direct computation from Lemma 7 and Eq. (22). □

We can easily specialize this result for the default weights. Recall that, in this case, $e_{j,b^\star_j} = 1$ and $c_j = c$ is a constant. Furthermore, $C = c^d$.

Corollary 12 (Computation of β, general f, default weights). *Assume that f is bounded on $\mathcal{S}$. Then it holds that*

$$\beta_0 = c^{-d}\left(1 + \sum_{j=1}^{d}\frac{1}{pc-1}\right)\mathbb{E}\left[\pi f(x)\right] - c^{-d}\sum_{j=1}^{d}\frac{pc}{pc-1}\mathbb{E}\left[\pi z_j f(x)\right] ,$$

and, for any $1 \leq j \leq d$,

$$\beta_j = c^{-d}\left(\frac{-pc}{pc-1}\mathbb{E}\left[\pi f(x)\right] + \frac{p^2c^2}{pc-1}\mathbb{E}\left[\pi z_j f(x)\right]\right) .$$

D.2 Additive f

We can specialize the results of the previous section when f has a specific structure. We begin with the case of an additive f.

Lemma 13 (Computation of β, additive f, general weights). *Assume that f satisfies Assumption 4.1. Then β is such that*

$$\beta_0 = \sum_{k=1}^{d} \frac{1}{pc_k - e_{k,b_k^\star}} \sum_{b \neq b_k^\star} e_{k,b}^{f_k} \,,$$

and, for any $1 \leq j \leq d$,

$$\beta_j = \frac{pc_j}{e_{j,b_j^\star}(pc_j - e_{j,b_j^\star})} \left(e_{j,b_j^\star}^{f_j} - \frac{e_{j,b_j^\star}}{pc_j} \sum_{b=1}^{p} e_{j,b}^{f_j} \right) .$$

Proof. First let us treat the case $j = 0$. We write

$$\begin{aligned}
\beta_0 &= \left(1 + \sum_{j=1}^{d} \frac{e_{j,b_j^\star}}{pc_j - e_{j,b_j^\star}} \right) \left(\sum_{k=1}^{d} \frac{1}{pc_k} \sum_{b=1}^{p} e_{k,b}^{f_k} \right) \\
&\quad - \sum_{j=1}^{d} \frac{pc_j}{pc_j - e_{j,b_j^\star}} \left(\sum_{k=1}^{p} \frac{e_{j,b_j^\star}}{p^2 c_j c_k} \sum_{b=1}^{p} e_{k,b}^{f_k} + \frac{1}{pc_j} \left[e_{j,b_j^\star}^{f_j} - \frac{e_{j,b_j^\star}}{pc_j} \sum_{b=1}^{p} e_{j,b}^{f_j} \right] \right) \\
&= \sum_{k=1}^{d} \frac{1}{pc_k} \sum_{b=1}^{p} e_{k,b_k^\star}^{f_k} - \sum_{j=1}^{d} \frac{1}{pc_j - e_{j,b_j^\star}} \left[e_{j,b_j^\star}^{f_j} - \frac{e_{j,b_j^\star}}{pc_j} \sum_{b=1}^{p} e_{j,b}^{f_j} \right]
\end{aligned}$$

We conclude after changing the indices in the sum and some algebra. As for the other terms,

$$\begin{aligned}
\beta_j &= \frac{-pc_j}{pc_j - e_{j,b_j^\star}} \sum_{k=1}^{d} \frac{1}{pc_k} \sum_{b=1}^{p} e_{k,b}^{f_k} \\
&\quad + \frac{p^2 c_j^2}{e_{j,b_j^\star}(pc_j - e_{j,b_j^\star})} \left[\sum_{k=1}^{d} \frac{e_{j,b_j^\star}}{p^2 c_j c_k} \sum_{b=1}^{p} e_{k,b}^{f_k} + \frac{1}{pc_j} \left(e_{j,b_j^\star}^{f_j} - \frac{e_{j,b_j^\star}}{pc_j} \sum_{b=1}^{p} e_{j,b}^{f_j} \right) \right]
\end{aligned}$$

and we obtain the promised result after some simplifications. □

We can specialize Lemma 13 even further if f is linear.

Corollary 14 (Computation of β, linear f, general weights). *If $f(x) = f_0 + f_1 x_1 + \cdots + f_d x_d$, we have*

$$\beta_0 = f(\gamma) - \sum_{j=1}^{d} \frac{1}{pc_j - e_{j,b_j^\star}} (e_{j,b_j^\star}^{x} - \gamma_j \cdot e_{j,b_j^\star}) f_j \,,$$

and, for any $1 \leq j \leq d$,

$$\frac{pc_j}{e_{j,b_j^\star}(pc_j - e_{j,b_j^\star})} (e_{j,b_j^\star}^{x} - \gamma_j \cdot e_{j,b_j^\star}) f_j \,.$$

Let us see how we can recover the analysis of Garreau and von Luxburg (2020) in the linear case.

Example D.1 (Computation of β, linear f, old analysis). We have seen before that, in this case, $\gamma = m$. Moreover,

$$pc_j - e_{j,b_j^\star} = (1-\alpha_j)\frac{p\nu}{\sqrt{\nu^2+\sigma^2}}\exp\left(\frac{-(\xi_j-\mu_j)^2}{2(\nu^2+\sigma^2)}\right).$$

Then we write

$$\begin{aligned} e^x_{j,b_j^\star} - \gamma_j e_{j,b_j^\star} &= (\alpha_j m_j - \theta_j)\frac{p\nu}{\sqrt{\nu^2+\sigma^2}}\exp\left(\frac{-(\xi_j-\mu_j)^2}{2(\nu^2+\sigma^2)}\right) \\ &\quad - m_j\alpha_j\frac{p\nu}{\sqrt{\nu^2+\sigma^2}}\exp\left(\frac{-(\xi_j-\mu_j)^2}{2(\nu^2+\sigma^2)}\right) \\ &= -\theta_j\frac{p\nu}{\sqrt{\nu^2+\sigma^2}}\exp\left(\frac{-(\xi_j-\mu_j)^2}{2(\nu^2+\sigma^2)}\right). \end{aligned}$$

We deduce that

$$\frac{e^x_{j,b_j^\star} - \gamma_j e_{j,b_j^\star}}{pc_j - e_{j,b_j^\star}} = \frac{-\theta_j}{1-\alpha_j}, \tag{24}$$

and therefore

$$\beta_0 = f(m) + \sum_{j=1}^d \frac{\theta_j f_j}{1-\alpha_j}.$$

Now, for any given $j > 0$, $pc_j/e_{j,b_j^\star} = 1/\alpha_j$. Combining with Eq. (24) we obtain

$$\beta_j = \frac{-\theta_j f_j}{\alpha_j(1-\alpha_j)}.$$

This is the expression of β appearing in Theorem 3.1 of Garreau and von Luxburg (2020).

D.3 Multiplicative f

When f is multiplicative (Assumption 4.2), we can also be more precise in the computation of β.

Lemma 15 (Computation of β, multiplicative f, general weights). *Assume that f satisfies Assumption 4.2 and is bounded on $\mathcal{S}$. Then*

$$\beta_0 = \frac{\prod_{k=1}^d c_k^{f_k}}{C}\left[1 + \sum_{j=1}^d \frac{e_{j,b_j^\star}}{pc_j - e_{j,b_j^\star}}\left(1 - \frac{e_{j,b_j^\star}^{f_j}}{e_{j,b_j^\star}}\cdot\frac{c_j}{c_j^{f_j}}\right)\right],$$

and, for any $1 \leq j \leq d$,

$$\beta_j = \frac{\prod_{k=1}^d c_k^{f_k}}{C}\cdot\frac{pc_j}{pc_j - e_{j,b_j^\star}}\left(\frac{e_{j,b_j^\star}^{f_j}}{e_{j,b_j^\star}}\cdot\frac{c_j}{c_j^{f_j}} - 1\right).$$

Proof. In view of Lemma 11, we just have to compute $\mathbb{E}[\pi f(x)]$ and $\mathbb{E}[\pi z_j f(x)]$ for any given $1 \leq j \leq d$. We begin with the computation of $\mathbb{E}[\pi f(x)]$:

$$\mathbb{E}[\pi f(x)] = \mathbb{E}\left[\prod_{k=1}^{k} \exp\left(\frac{-(\tau_k(x_k) - \tau_k(\xi_k))^2}{2\nu^2}\right) f_k(x_{ik})\right] \quad \text{(Assumption 4.2 + Eq. (16))}$$

$$= \prod_{k=1}^{d} \mathbb{E}\left[\exp\left(\frac{-(\tau_k(x_k) - \tau_k(\xi_k))^2}{2\nu^2}\right) f_k(x_{ik})\right] \quad \text{(independence)}$$

$$\mathbb{E}[\pi f(x)] = \prod_{k=1}^{d} c_k^{f_k} \quad \text{(Lemma 26)}$$

The second computation is very similar in spirit:

$$\mathbb{E}[\pi z_j f(x)] = \mathbb{E}\left[\prod_{\substack{k=1 \\ k\neq j}}^{k} \mathrm{e}^{\frac{-(\tau_k(x_k)-\tau_k(\xi_k))^2}{2\nu^2}} f_k(x_k) \cdot \mathrm{e}^{\frac{-(\tau_j(x_j)-\tau_j(\xi_j))^2}{2\nu^2}} z_j f_j(x_j)\right]$$

(Assumption 4.2 + Eq. (16))

$$= \prod_{\substack{k=1 \\ \neq j}}^{d} \mathbb{E}\left[\exp\left(\frac{-(\tau_k(x_k) - \tau_k(\xi_k))^2}{2\nu^2}\right) f_k(x_k)\right] \cdot \mathbb{E}\left[\mathrm{e}^{\frac{-(\tau_j(x_j)-\tau_j(\xi_j))^2}{2\nu^2}} z_j f_j(x_j)\right]$$

(independence)

$$\mathbb{E}[\pi z_j f(x)] = \prod_{\substack{k=1 \\ k\neq j}}^{d} c_k^{f_k} \cdot \frac{e_{j,b_j^\star}^{f_j}}{p}\,. \quad \text{(Lemma 26)}$$

Simple algebra concludes the proof. □

Appendix E. Proof of Lemma 2

In this Appendix, we prove Lemma 2.

Proof. First let us set $h := f - g$ and $\varepsilon := \|h\|_\infty$. We notice that

$$\|\beta^f - \beta^g\| = \|\Sigma^{-1}\Gamma^f - \Sigma^{-1}\Gamma^g\| = \|\Sigma^{-1}\Gamma^h\|\,.$$

Let us focus first on the first coordinate:

$$(\Sigma^{-1}\Gamma^h)_0 = C^{-1}\left(1 + \sum_{j=1}^{d} \frac{1}{pc-1}\right)\mathbb{E}[\pi h(x)] + C^{-1}\sum_{j=1}^{d}\frac{-pc}{pc-1}\mathbb{E}[\pi z_j h(x)]$$

$$= C^{-1}\,\mathbb{E}[\pi h(x)] + C^{-1}\sum_{j=1}^{d}\frac{1}{pc-1}\mathbb{E}[\pi(1-pcz_j)h(x)]\,.$$

Recall Lemma 28:

$$\left|C^{-1}\,\mathbb{E}[\pi h(x)]\right| \leq \varepsilon\,.$$

As for the second part, we write

$$\begin{aligned}\mathbb{E}\left[\pi\left|1-pcz_j\right|h(x)\right] &\leq (pc\,\mathbb{E}\left[\pi z_j\right]+\mathbb{E}\left[\pi\right])\varepsilon \\ &= (pc\cdot\frac{C}{pc}+C)\varepsilon \qquad \text{(Lemma 28)} \\ \mathbb{E}\left[\pi\left|1-pcz_j\right|h(x)\right] &\leq 2C\varepsilon\end{aligned}$$

We deduce

$$\left|(\Sigma^{-1}\Gamma^h)_0\right| \leq \left(1+\frac{2d}{p-1}\,\mathrm{e}^{\frac{1}{2\nu^2}}\right)\varepsilon\,. \tag{25}$$

Now let us set $j \geq 1$.

$$\begin{aligned}(\Sigma^{-1}\Gamma^h)_j &= C^{-1}\left[\frac{-pc}{pc-1}\,\mathbb{E}\left[\pi h(x)\right]+\frac{p^2c^2}{pc-1}\,\mathbb{E}\left[\pi z_j h(x)\right]\right] \\ &= \frac{C^{-1}pc}{pc-1}\,\mathbb{E}\left[\pi(pcz_j-1)h(x)\right]\,.\end{aligned}$$

As before, we obtain

$$\left|(\Sigma^{-1}\Gamma^h)_j\right| \leq \frac{2pc\varepsilon}{pc-1} \leq \frac{2p\,\mathrm{e}^{\frac{1}{2\nu^2}}\,\varepsilon}{p-1}\,. \tag{26}$$

We then collect Eq. (25) and (26) to obtain the promised bound. □

Appendix F. Concentration of $\hat{\Sigma}_n$

In this section, we show that $\hat{\Sigma}_n$ is concentrated around Σ in operator norm. The idea is to use standard results on the concentration of sum of matrices (Vershynin, 2018). Indeed, $\hat{\Sigma}$ can be written as $\frac{1}{n}\sum_{i=1}^n \pi_i Z_i Z_i^\top$. Since each of these matrices are bounded and identically distributed, we turn to a Hoeffding-type inequality. We borrow the following result from Tropp (2012).

Theorem 16 (Matrix Hoeffding (Tropp, 2012)). *Consider a finite sequence M_i of independent, random, symmetric matrices with dimension D, and let A_i be a sequence of fixed symmetric matrices. Assume that each random matrix satisfies*

$$\mathbb{E}\left[M_i\right] = 0 \quad \textit{and} \quad M_i^2 \preccurlyeq A_i^2 \quad \textit{almost surely}.$$

Then, for all $t \geq 0$,

$$\mathbb{P}\left(\lambda_{\max}\left(\sum_{i=1}^n M_i\right) \geq t\right) \leq D\cdot\exp\left(\frac{-t^2}{8\sigma^2}\right)\,,$$

where $\sigma^2 := \left\|\sum_{i=1}^n A_i^2\right\|_{\mathrm{op}}$.

We slightly adapt this result for the situation at hand.

Corollary 17 (Matrix Hoeffding, bounded entries). *Consider a finite sequence M_i of independent, centered, random, symmetric matrices with dimension D. Assume that the entries of each matrix satisfy*

$$(M_i)_{j,k} \in [-1, 1] \quad \textit{almost surely}.$$

Then, for all $t \geq 0$,

$$\mathbb{P}\left(\left\| \frac{1}{n} \sum_{i=1}^{n} M_i \right\|_{\text{op}} \geq t \right) \leq 2D \cdot \exp\left(\frac{-nt^2}{8D^2} \right) .$$

Proof. Let $u \in \mathbb{R}^D$ such that $\|u\| = 1$. We write

$$\begin{aligned} |(Mi_u)_j| &= \sum_{k=1}^{D} (M_i)_{j,k} u_k \\ &\leq \|(M_i)_j\| \cdot \|u\| && \text{(Cauchy-Schwarz)} \\ &\leq \sqrt{D}\,. && \text{(bounded a.s. + } \|u\| = 1) \end{aligned}$$

We deduce that $\|M_i u\| \leq D$ almost surely. Since we considered an arbitrary u, we have showed that $\|M_i\|_{\text{op}} \leq D$ almost surely for any i. Thus we can apply Th. 16 with $A_i = D\,\mathrm{I}_D$ to obtain

$$\mathbb{P}\left(\lambda_{\max}\left(\sum_{i=1}^{n} M_i \right) \geq t \right) \leq D \cdot \exp\left(\frac{-t^2}{8nD^2} \right) ,$$

using the fact that A_i commutes with M_i and thus $M_i \preccurlyeq A_i$ implies $M_i^2 \preccurlyeq A_i^2$. The result is a direct consequence from the last display. □

We can now proceed to the main result of this section, the concentration of $\hat{\Sigma}$ around its mean Σ.

Lemma 18 (Concentration of $\hat{\Sigma}$, general weights). *For any $t \geq 0$,*

$$\mathbb{P}\left(\|\hat{\Sigma} - \Sigma\|_{\text{op}} \geq t \right) \leq 4d \cdot \exp\left(\frac{-nt^2}{32d^2} \right) .$$

Proof. Recall Eq. (19): the entries of $\pi_i Z_i Z_i^\top$ belong to $[0, 1]$ almost surely since $\pi_i \in [0, 1]$ for any weights satisfying Eq. (16) and $z_{ij} \in [0, 1]$. As a consequence, so do the entries of Σ. Let us set

$$M_i := \pi_i Z_i Z_i^\top - \Sigma\,.$$

Then M_i satisfies the assumptions of Th. 17 with $D = d + 1$ and the result follows since $\frac{1}{n}\sum_{i=1}^{n} M_i = \hat{\Sigma} - \Sigma$. □

Appendix G. Concentration of $\hat{\Gamma}$

The goal of this section is the concentration of $\hat{\Gamma}$. Interestingly, we could not find a multivariate version of Hoeffding's inequality (Vershynin, 2018). We resort to a combination of Hoeffding's inequality in the univariate case and a union bound argument.

Theorem 19 (Hoeffding's inequality). *Let M_i be a finite sequence of centered random variables such that*

$$M_i \in [-M, M] \quad \textit{almost surely}.$$

Then, for any $t \geq 0$,

$$\mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^{n} M_i\right| \geq t\right) \leq 2 \cdot \exp\left(\frac{-nt^2}{2M^2}\right).$$

Proof. This is Th. 2.8 in Boucheron et al. (2013) in our notation. □

Lemma 20 (Concentration of $\hat{\Gamma}$, general weights). *Suppose that f is bounded by M on $\mathcal{S}$. Then, for any $t \geq 0$,*

$$\mathbb{P}\left(\|\hat{\Gamma} - \Gamma\| \geq t\right) \leq 4d \exp\left(\frac{-nt^2}{32Md^2}\right).$$

Proof. The components of $\hat{\Gamma}$ are given by $\pi_i f(x_i)$ and $\pi_i z_{i,j} f(x_i)$. Since f is bounded by M and $\pi_i, z_{i,j} \in [0,1]$ for any weights satisfying Eq. (16), these quantities live in $[0, M]$. We deduce that, for any i, j,

$$\begin{cases} \pi_i f(x_i) - \Gamma_0 \in [-2M, 2M] \\ \pi_i z_{i,j} f(x_i) - \Gamma_j \in [-2M, 2M] \end{cases}$$

almost surely. We can apply Th. 19 coordinate by coordinate: for any $t \geq 0$,

$$\begin{cases} \mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^{n} \pi_i f(x_i) - \Gamma_0\right| \geq t\right) \leq 2 \cdot \exp\left(\frac{-nt^2}{8M^2}\right) \\ \mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^{n} \pi_i z_{i,j} f(x_i) - \Gamma_j\right| \geq t\right) \leq 2 \cdot \exp\left(\frac{-nt^2}{8M^2}\right) \end{cases}$$

By a union bound argument,

$$\mathbb{P}\left(\|u\| \geq t\right) \leq \mathbb{P}\left(\max_i |u_i| \geq t/D\right) \leq D \cdot \mathbb{P}\left(|u_i| \geq t/D\right),$$

we deduce the result. (Note that, as usual, we used $d + 1 \leq 2d$.) □

Appendix H. Proof of the main result

In this section we prove that $\hat{\beta}$ is concentrated around β.

H.1 Binding lemma

We begin by a somewhat technical result, showing that we can control the behavior of $\|\hat{\beta} - \Sigma^{-1}\Gamma\|$ by controlling $\|\hat{\Sigma} - \Sigma\|_{\mathrm{op}}$, $\|\hat{\Gamma} - \Gamma\|$, and $\left\|\Sigma^{-1}\right\|_{\mathrm{op}}$.

Lemma 21 (Control of $(\mathrm{I}_D + H)^{-1}$). *Let $H \in \mathbb{R}^{D\times D}$ be a matrix such that $\|H\|_{\mathrm{op}} \leq -1 + \frac{\sqrt{7}}{2} (\approx 0.32)$. Then $\mathrm{I}_D + H$ is invertible and*

$$\left\|(\mathrm{I}_D + H)^{-1}\right\|_{\mathrm{op}} \leq 2\,.$$

Proof. We first show that $\mathrm{I}_D + H$ is invertible. Indeed, suppose that $\mathrm{Ker}\,(\mathrm{I}_D + H) \neq \{0\}$. Then, in particular, there exists $x \in \mathbb{R}^D$ with unit norm such that $Hx = -x$. Since $\|H\|_{\mathrm{op}} = \max_{\|x\|=1} \|Hx\|$, we would have $\|H\|_{\mathrm{op}} \geq 1$, which contradicts $\|H\|_{\mathrm{op}} \leq 0.32$.

Now according to Lemma 30,

$$\left\|(\mathrm{I}_D + H)^{-1}\right\|_{\mathrm{op}} = (\lambda_{\min}\left((\mathrm{I}_D + H)^\top(\mathrm{I}_D + H)\right))^{-1/2}\,. \tag{27}$$

Let us set $M := (\mathrm{I}_D + H)^\top(\mathrm{I}_D + H)$ and find a lower bound on $\lambda_{\min}(M)$. We first notice that M is positive semi-definite. Thus we now that $\lambda_{\min}(M) = \min_{\|x\|=1} x^\top M x$. Let us fix $x \in S^{D-1}$ for now. Then

$$x^\top M x = x^\top(M - \mathrm{I}_D)x + x^\top x = x^\top(M - \mathrm{I}_D)x + 1\,,$$

since $\|x\| = 1$. But on the other side,

$$\begin{aligned} -x^\top(M - \mathrm{I}_D)x &\leq |\langle x, (M - \mathrm{I}_D)x\rangle| \\ &\leq \|x\| \cdot \|(M - \mathrm{I}_D)x\| && \text{(Cauchy-Schwarz)} \\ &\leq \|M - \mathrm{I}_D\|_{\mathrm{op}} && \text{(definition of } \|\cdot\|_{\mathrm{op}} + \|x\| = 1) \end{aligned}$$

Moreover, by the triangle inequality and sub-multiplicativity of the operator norm,

$$\|M - \mathrm{I}_D\|_{\mathrm{op}} = \left\|H + H^\top + H^\top H\right\|_{\mathrm{op}} \leq 2\|H\|_{\mathrm{op}} + \|H\|_{\mathrm{op}}^2\,.$$

We deduce that

$$\lambda_{\min}(M) \geq 1 - 2\|H\|_{\mathrm{op}} - \|H\|_{\mathrm{op}}^2\,, \tag{28}$$

which is a positive quantity since we assumed $\|H\|_{\mathrm{op}} \leq 0.32$.

Plugging the lower bound (28) into Eq. (27), we obtain

$$\left\|(\mathrm{I}_D + H)^{-1}\right\|_{\mathrm{op}} \leq \frac{1}{\sqrt{1 - 2\|H\|_{\mathrm{op}} - \|H\|_{\mathrm{op}}^2}}\,.$$

Set $\psi(x) := (1 - 2x - x^2)^{-1/2}$. One can easily check that $\psi(x) \leq 2$ for any $x \in [0, -1 + \sqrt{7}/2]$. □

Remark H.1. The numerical constant 2 in the statement of Lemma 21 can be replaced by any arbitrary constant greater than 1, at the cost of constraining further the range of $\|H\|_{\mathrm{op}}$.

Remark H.2. The Hoffman-Wielandt inequality also yields a lower bound on $\lambda_{\min}(M)$, giving essentially the same result but with the Frobenius norm instead of the operator norm. Since we know how to control $\left\|\hat{\Sigma}-\Sigma\right\|_{\text{op}}$, we prefer this version of the result.

Using Lemma 21 we can prove something more interesting.

Lemma 22 (Control of $(\mathrm{I}_D+H)^{-1}-\mathrm{I}_d$). *Let $H\in\mathbb{R}^{D\times D}$ be a matrix such that $\|H\|_{\text{op}}\leq -1+\frac{\sqrt{7}}{2}(\approx 0.32)$. Then I_D+H is invertible, and*

$$\left\|(\mathrm{I}_D+H)^{-1}-\mathrm{I}_D\right\|_{\text{op}}\leq 2\,\|H\|_{\text{op}}\,.$$

Proof. According to Lemma 21, I_D+H is an invertible matrix. Now we write

$$(\mathrm{I}_D+H)^{-1}-\mathrm{I}_D=(\mathrm{I}_D+H)^{-1}(\mathrm{I}_D-(\mathrm{I}_D+H))=-(\mathrm{I}_D+H)^{-1}H\,. \tag{29}$$

Since the operator norm is sub-multiplicative, Eq. (29) implies that

$$\left\|(\mathrm{I}_D+H)^{-1}-\mathrm{I}_D\right\|_{\text{op}}\leq\left\|(\mathrm{I}_D+H)^{-1}\right\|_{\text{op}}\cdot\|H\|_{\text{op}}\,.$$

and Lemma 21 guarantees that $\left\|(\mathrm{I}_D+H)^{-1}\right\|_{\text{op}}\leq 2$ under our assumptions. □

Remark H.3. It is a good surprise that the constants in Lemma 22 do not depend on the dimension. In fact, we believe that this result is nearly optimal. Indeed, in the unidimensional case, one can show that

$$\left|\frac{1}{1+h}-1\right|\leq 2\,|h|\,,$$

for any $h\in\mathbb{R}$ such that $|h|\leq\frac{1}{2}$, showing that the inequality cannot be significantly improved.

We are now able to state and prove the main result of this section.

Lemma 23 (Binding lemma). *Let $X\in\mathbb{R}^{D\times D}$ such that X is invertible and $Y\in\mathbb{R}^D$. Then, for any $H\in\mathbb{R}^{D\times D}$ such that $\left\|X^{-1}H\right\|_{\text{op}}\leq 0.32$ and any $H'\in\mathbb{R}^D$, it holds that*

$$\left\|(X+H)^{-1}(Y+H')-X^{-1}Y\right\|\leq 2\left\|X^{-1}\right\|_{\text{op}}\left\|H'\right\|+2\left\|X^{-1}\right\|_{\text{op}}^2\|Y\|\,\|H\|_{\text{op}}\,. \tag{30}$$

In particular, we achieve the promised control by setting $X=\Sigma$, $Y=\Gamma$, $H=\hat{\Sigma}-\Sigma$, and $H'=\hat{\Gamma}-\Gamma$ in Lemma 23. Namely,

$$\left\|\hat{\beta}-\beta\right\|\leq 2\left\|\Sigma^{-1}\right\|_{\text{op}}\left\|\hat{\Gamma}-\Gamma\right\|+2\left\|\Sigma^{-1}\right\|_{\text{op}}^2\|\Gamma\|\left\|\hat{\Sigma}-\Sigma\right\|_{\text{op}}\,. \tag{31}$$

Proof. We first notice that since $\left\|X^{-1}H\right\|_{\text{op}}\leq 0.32$, the matrix $\mathrm{I}_D+X^{-1}H$ is invertible according to Lemma 21. We deduce that $X+H$ is also invertible, with

$$(X+H)^{-1}=\big(X(\mathrm{I}_D+X^{-1}H)\big)^{-1}=(\mathrm{I}_D+X^{-1}H)^{-1}X^{-1}\,. \tag{32}$$

Let us split the left-hand side of Eq. (30): by the triangle inequality,

$$\begin{aligned}\left\|(X+H)^{-1}(Y+H') - X^{-1}Y\right\| &\leq \left\|(X+H)^{-1}(Y+H') - (X+H)^{-1}Y\right\| \\ &\quad + \left\|(X+H)^{-1}Y - X^{-1}Y\right\| \\ &= \left\|(X+H)^{-1}H'\right\| + \left\|(X+H)^{-1}Y - X^{-1}Y\right\| .\end{aligned}$$

Let us focus on the first term. We write

$$\begin{aligned}\left\|(X+H)^{-1}\right\|_{\mathrm{op}} &= \left\|(\mathrm{I}_D + X^{-1}H)^{-1}X^{-1}\right\|_{\mathrm{op}} && \text{(Eq. 32)} \\ &\leq \left\|(\mathrm{I}_D + X^{-1}H)^{-1}\right\|_{\mathrm{op}} \cdot \left\|X^{-1}\right\|_{\mathrm{op}} && (\|\cdot\|_{\mathrm{op}} \text{ is sub-multiplicative}) \\ &\leq 2\left\|X^{-1}\right\|_{\mathrm{op}} . && \text{(Lemma 21)}\end{aligned}$$

From the last display we deduce that

$$\left\|(X+H)^{-1}H'\right\| \leq 2\left\|X^{-1}\right\|_{\mathrm{op}}\left\|H'\right\| . \tag{33}$$

Now for the second term, we have

$$\begin{aligned}\left\|(X+H)^{-1} - X^{-1}\right\|_{\mathrm{op}} &= \left\|(\mathrm{I}_D + X^{-1}H)^{-1}X^{-1} - X^{-1}\right\|_{\mathrm{op}} && \text{(Eq. (32))} \\ &\leq \left\|(\mathrm{I}_D + X^{-1}H)^{-1} - \mathrm{I}_d\right\|_{\mathrm{op}} \cdot \left\|X^{-1}\right\|_{\mathrm{op}} \\ & && (\|\cdot\|_{\mathrm{op}} \text{ is sub-multiplicative}) \\ &\leq 2\left\|X^{-1}H\right\|_{\mathrm{op}} \cdot \left\|X^{-1}\right\|_{\mathrm{op}} && \text{(Lemma 22)} \\ \left\|(X+H)^{-1} - X^{-1}\right\|_{\mathrm{op}} &\leq 2\left\|H\right\|_{\mathrm{op}} \cdot \left\|X^{-1}\right\|_{\mathrm{op}}^2 && (\|\cdot\|_{\mathrm{op}} \text{ is sub-multiplicative})\end{aligned}$$

We deduce that

$$\left\|(X+H)^{-1}Y - X^{-1}Y\right\| \leq 2\left\|H\right\|_{\mathrm{op}} \cdot \left\|X^{-1}\right\|_{\mathrm{op}}^2 \cdot \|Y\| . \tag{34}$$

We conclude the proof by adding Eq. (33) and Eq. (34). □

H.2 Concentration of $\hat{\beta}_n$

We are now able to state and prove our main result, the concentration of $\hat{\beta}_n$ around β for a general f and arbitrary weights satisfying Eq. 16.

Theorem 24 (Concentration of $\hat{\beta}_n$, general f, general weights). *Suppose that f is bounded by M on $\mathcal{S}$. Let $\varepsilon > 0$ be a small constant , at least smaller than $1/M$. Let $\eta \in (0,1)$. Then, for every*

$$n \geq \max\left\{\frac{2^{12}Md^4p^4\log \mathrm{e}^{\frac{4}{\nu^2}}\frac{8d}{\eta}}{C^2\varepsilon^2}, \frac{2^{15}M^2d^7p^8\,\mathrm{e}^{\frac{8}{\nu^2}}\frac{8d}{\eta}}{C^4\varepsilon^2}\right\},$$

we have $\mathbb{P}\left(\left\|\hat{\beta}_n - \beta\right\| \geq \varepsilon\right) \leq \eta$.

Proof. Let us define

$$t_1 := \frac{C\varepsilon}{8\sqrt{2}dp^2\,\mathrm{e}^{\frac{2}{\nu^2}}} \quad \text{and} \quad t_2 := \frac{C^2\varepsilon}{32Md^{5/2}p^4\,\mathrm{e}^{\frac{4}{\nu^2}}}\,.$$

According to Lemma 20 and 18, we can build events Ω_1 and Ω_2 such that: (i) on Ω_1, $\left\|\hat{\Gamma}_n - \Gamma\right\| \leq t_1$ with probability higher than $1 - 4d\exp\left(-nt_1^2/(32Md^2)\right)$, and (ii) on Ω_2, $\left\|\hat{\Sigma}_n - \Sigma\right\|_{\mathrm{op}} \leq t_2$ with probability higher than $1 - 4d\exp\left(-nt_2^2/(32d^2)\right)$. Let us define

$$n_1 := \frac{2^{12}Md^4p^4\log\mathrm{e}^{\frac{4}{\nu^2}}\,\frac{8d}{\eta}}{C^2\varepsilon^2} \quad \text{and} \quad n_2 := \frac{2^{15}M^2d^7p^8\,\mathrm{e}^{\frac{8}{\nu^2}}\,\frac{8d}{\eta}}{C^4\varepsilon^2}\,.$$

By assumption, n is larger than $\max(n_1, n_2)$. One can check that, in this case, Ω_1 and Ω_2 both have probability higher than $1-\eta/2$. We now work on the event $\Omega := \Omega_1 \cap \Omega_2$. By the union bound, Ω has probability greater than $1-\eta$. Let us show that, on Ω, $\left\|\hat{\beta}_n - \beta\right\| \leq \varepsilon$.

First note that, according to Lemma 8, $\left\|\Sigma^{-1}\right\|_{\mathrm{op}} \leq \frac{2sqrt2dp^2\,\mathrm{e}^{\frac{1}{2\nu^2}}}{C}$. Thus, since the operator norm is sub-multiplicative, we have

$$\begin{aligned}
\left\|\Sigma^{-1}(\hat{\Sigma}_n - \Sigma)\right\|_{\mathrm{op}} &\leq \left\|\Sigma^{-1}\right\|_{\mathrm{op}} \left\|\hat{\Sigma}_n - \Sigma\right\|_{\mathrm{op}} \\
&\leq \frac{2\sqrt{2}dp^2\,\mathrm{e}^{\frac{1}{2\nu^2}}}{C} \cdot \frac{C^2\varepsilon}{32Md^{5/2}p^4\,\mathrm{e}^{\frac{4}{\nu^2}}} \\
&\leq \frac{\sqrt{2}}{16} \cdot \frac{\varepsilon}{M} \cdot C \cdot \frac{1}{d^{3/2}} \cdot \frac{1}{p^2} \cdot \mathrm{e}^{\frac{-2}{\nu^2}}\,.
\end{aligned}$$

Since we assumed $\varepsilon < M$, $\left\|\Sigma^{-1}(\hat{\Sigma}_n - \Sigma)\right\|_{\mathrm{op}} \leq \sqrt{2}/16 < 0.32$. Therefore we can use Eq. (31) and the result follows. □

Appendix I. Extension of Lemma 5

We now present a generalization and a proof of Lemma 5 for general weights.

Lemma 25 (Ignoring unused coordinates, general weights). *Assume that f satisfies Assumption 4.3 and is bounded on $\mathcal{S}$. Let $j \in \overline{S}$, where S is the set of indices relevant for f and $\overline{S} := \{1, \ldots, d\} \setminus S$. Then $\beta_j = 0$.*

Proof. The proof is a direct application of Theorem 24, and can be seen as a straightforward generalization of the proof of Lemma 5. We compute first

$$\begin{aligned}
\mathbb{E}\left[\pi f(x)\right] &= \mathbb{E}\left[\prod_{k=1}^{d} \mathrm{e}^{\frac{-(\tau_k(x_k)-\tau_k(\xi_k))^2}{2\nu^2}} f(x)\right] && \text{(Eq. (16))}\\
&= \mathbb{E}\left[\prod_{k=1}^{d} \mathrm{e}^{\frac{-(\tau_k(x_k)-\tau_k(\xi_k))^2}{2\nu^2}} g(x_{j_1},\dots,x_{j_s})\right] && \text{(Assumption 4.3)}\\
&= \prod_{k\in\overline{S}} \mathbb{E}\left[\mathrm{e}^{\frac{-(\tau_k(x_k)-\tau_k(\xi_k))^2}{2\nu^2}}\right] \cdot \mathbb{E}\left[\prod_{k\in S} \mathrm{e}^{\frac{-(\tau_k(x_k)-\tau_k(\xi_k))^2}{2\nu^2}} g(x_{j_1},\dots,x_{j_s})\right] && \text{(independence)}\\
\mathbb{E}\left[\pi f(x)\right] &= \prod_{k\in\overline{S}} c_k \cdot G\,, && \text{(Lemma 26)}
\end{aligned}$$

where we set

$$G := \mathbb{E}\left[\prod_{k\in S} \mathrm{e}^{\frac{-(\tau_k(x_k)-\tau_k(\xi_k))^2}{2\nu^2}} g(x_{j_1},\dots,x_{j_s})\right]$$

in the last display. The other computation is similar. Recall that $j \notin S$:

$$\begin{aligned}
\mathbb{E}\left[\pi z_j f(x)\right] &= \mathbb{E}\left[\prod_{k=1}^{d} \mathrm{e}^{\frac{-(\tau_k(x_k)-\tau_k(\xi_k))^2}{2\nu^2}} z_j f(x)\right] && \text{(Eq. (16))}\\
&= \prod_{k\in\overline{S}\setminus\{j\}} c_k \cdot \mathbb{E}\left[\mathrm{e}^{\frac{-(\tau_j(x_j)-\tau_j(\xi_j))^2}{2\nu^2}} z_j\right] \cdot \mathbb{E}\left[\prod_{k\in S} \mathrm{e}^{\frac{-(\tau_k(x_k)-\tau_k(\xi_k))^2}{2\nu^2}} g(x_{j_1},\dots,x_{j_s})\right] && \text{(independence)}\\
\mathbb{E}\left[\pi z_j f(x)\right] &= \frac{\prod_{k\in\overline{S}} c_k}{c_j} \cdot \frac{1}{p} e_{j,b_j^\star} \cdot G\,.
\end{aligned}$$

Finally we write

$$\begin{aligned}
\beta_j &= C^{-1} \frac{pc_j}{pc_j - e_{j,b_j^\star}} \left(-\mathbb{E}\left[\pi f(x)\right] + \frac{pc_j}{e_{j,b_j^\star}} \mathbb{E}\left[\pi z_j f(x)\right]\right) && \text{(Lemma 11)}\\
&= C^{-1} \frac{pc_j}{pc_j - e_{j,b_j^\star}} \left(-\prod_{k\in\overline{S}} c_k \cdot G + \frac{pc_j}{e_{j,b_j^\star}} \frac{\prod_{k\in\overline{S}} c_k}{c_j} \cdot \frac{1}{p} e_{j,b_j^\star} \cdot G\right)\\
\beta_j &= 0\,.
\end{aligned}$$

□

Appendix J. Technical results

J.1 Expectation computations

In this section we collect the expected values computation needed for the computation of Σ and Γ. We begin with a generic lemma, a very common computation in our proofs which

appears each time we use the independence assumption between the x_{ij} and we split the π_i product.

Lemma 26 (Basic computation). *Let $\psi : \mathbb{R} \to \mathbb{R}$ be a bounded function on the support of x_j for any $1 \leq j \leq d$. Then*

$$\begin{cases} \mathbb{E}\left[\exp\left(\frac{-(\tau_j(x_j)-\tau_j(\xi_j))^2}{2\nu^2}\right)\psi(x_j)\right] = \frac{1}{p}\sum_{b=1}^{p} e_{j,b}^{\psi} \\ \mathbb{E}\left[\exp\left(\frac{-(\tau_j(x_j)-\tau_j(\xi_j))^2}{2\nu^2}\right)z_j\psi(x_j)\right] = \frac{1}{p}e_{j,b_j^\star}^{\psi}\,. \end{cases}$$

In particular,

$$\mathbb{E}\left[\exp\left(\frac{-1}{2\nu^2}(\tau_j(\xi_j)-\tau_j(x_j))^2\right)\right] = c_j\,. \tag{35}$$

Proof. Law of total expectation + definition of the $e_{j,b}$ coefficients. □

Lemma 26 is the reason why the $e_{j,b}$ are ubiquitous in our results. If the weights have some multiplicative structure, it is easy to extend Lemma 26 to the full weights, which we achieve in our next result.

Lemma 27 (Key computation). *Suppose that π_i satisfies Eq. (16). Let $\psi : \mathbb{R} \to \mathbb{R}$ be a function bounded on the support of x_j for any $1 \leq j \leq d$. Then, for any given i, j, k with $j \neq k$,*

$$\begin{cases} \mathbb{E}\left[\pi\psi(x_j)\right] = \frac{C}{pc_j}\sum_{b=1}^{p} e_{j,b}^{\psi} \\ \mathbb{E}\left[\pi z_j\psi(x_j)\right] = \frac{C}{pc_j}e_{j,b_j^\star}^{\psi} \\ \mathbb{E}\left[\pi z_j\psi(x_k)\right] = \frac{C}{p^2c_jc_k}e_{j,b_j^\star}\sum_{b=1}^{p} e_{k,b}^{\psi} \end{cases}$$

Proof. We write

$$\begin{aligned} \mathbb{E}\left[\pi\psi(x_j)\right] &= \mathbb{E}\left[\exp\left(\frac{-1}{2\nu^2}\sum_{k=1}^{d}(\tau_k(\xi_k)-\tau_k(x_k))^2\right)\cdot\psi(x_j)\right] && \text{(Eq. (16))} \\ &= \prod_{k\neq j} c_k \cdot \mathbb{E}\left[\psi(x_j)\exp\left(\frac{-1}{2\nu^2}(\tau_j(\xi_j)-\tau_j(x_j))^2\right)\right] && \text{(independence + Eq. (35))} \\ &= \prod_{k\neq j} c_k \cdot \frac{1}{p}\sum_{b=1}^{p} e_{j,b}^{\psi} && \text{(Lemma 26)} \\ \mathbb{E}\left[\pi\psi(x_j)\right] &= \frac{C}{pc_j}\sum_{b=1}^{p} e_{j,b}^{\psi}\,. && \text{(definition of } C\text{)} \end{aligned}$$

The proofs of the remaining results are quite similar:

$$\begin{aligned}
\mathbb{E}\left[\pi z_j \psi(x_j)\right] &= \mathbb{E}\left[\exp\left(\frac{-1}{2\nu^2}\sum_{k=1}^{d}(\tau_k(\xi_k)-\tau_k(x_k))^2\right)\cdot z_j\psi(x_j)\right] && \text{(Eq. (16))}\\
&= \prod_{k\neq j} c_k \cdot \mathbb{E}\left[\psi(x_j) z_j \exp\left(\frac{-1}{2\nu^2}(\tau_j(\xi_j)-\tau_j(x_{x_j}))^2\right)\right] \\
& && \text{(independence + Eq. (35))}\\
&= \prod_{k\neq j} c_k \cdot \frac{1}{p} e^{\psi}_{j,b_j^\star} && \text{(Lemma 26)}\\
\mathbb{E}\left[\pi z_j \psi(x_j)\right] &= \frac{C}{pc_j} e^{\psi}_{j,b_j^\star}
\end{aligned}$$

$$\begin{aligned}
\mathbb{E}\left[\pi z_j \psi(x_k)\right] &= \mathbb{E}\left[\exp\left(\frac{-1}{2\nu^2}\sum_{\ell=1}^{d}(\tau_\ell(\xi_\ell)-\tau_\ell(x_\ell))^2\right)\cdot z_j\psi(x_k)\right] && \text{(Eq. (16))}\\
&= \prod_{\ell\neq j,k} c_\ell \cdot \mathbb{E}\left[z_j \exp\left(\frac{-1}{2\nu^2}(\tau_j(\xi_j)-\tau_j(x_j))^2\right)\right] \\
&\qquad \cdot \mathbb{E}\left[\psi(x_k)\exp\left(\frac{-1}{2\nu^2}(\tau_k(\xi_k)-\tau_k(x_k))^2\right)\right] && \text{(independence)}\\
&= \prod_{\ell\neq j,k} c_\ell \cdot \frac{e_{j,b_j^\star}}{p}\cdot\frac{1}{p}\sum_{b=1}^{p} e^{\psi}_{k,b} && \text{(Lemma 26)}\\
\mathbb{E}\left[\pi z_j \psi(x_k)\right] &= \frac{C}{p^2 c_j c_k} e_{j,b_j^\star}\sum_{b=1}^{p} e^{\psi}_{k,b}\,. && \text{(definition of } C\text{)}
\end{aligned}$$

□

We specialize Lemma 27 in the case $\psi = 1$, since the $e_{j,b}$ coefficients are ubiquitous in our computations.

Lemma 28 (Expected values computations, zero-th order). *For any $j \neq k$,*

$$\begin{cases}
\mathbb{E}\left[\pi\right] = C\\
\mathbb{E}\left[\pi z_j\right] = C\frac{e_{j,b_j^\star}}{pc_j}\\
\mathbb{E}\left[\pi z_j z_k\right] = C\frac{e_{j,b_j^\star}}{pc_j}\frac{e_{k,b_k^\star}}{pc_k}
\end{cases}$$

Proof. The first two results are a direct consequence of Lemma (27) for $\psi = 1$. For the third one, we set $\psi(x) = \mathbb{1}_{x\in[q_{k,b_k^\star-1},q_{k,b_k^\star}]}$, and we notice that, in this case,

$$e^{\psi}_{k,b} = \begin{cases} e_{k,b_k^\star} \text{ if } b = b_k^\star \\ 0 \text{ otherwise.}\end{cases}$$

□

The case $\psi = \mathrm{id}$ is also of some importance in our analysis, let us specialize Lemma 27 in that case as well.

Lemma 29 (Expected values, first order). *Let $j, k \in \{1, \dots, d\}$ be fixed indices, with $j \neq k$. Then*

$$\begin{cases} \mathbb{E}\left[\pi x_j\right] = \frac{C}{pc_j} \sum_{b=1}^{p} e^x_{j,b} \\ \mathbb{E}\left[\pi z_j x_j\right] = \frac{C}{pc_j} e^x_{j,b^\star_j} \\ \mathbb{E}\left[\pi z_j x_k\right] = \frac{C}{p^2 c_j c_k} e_{j,b^\star_j} \sum_{b=1}^{p} e^x_{k,b} \end{cases}$$

Proof. Straightforward from Lemma 27 with $\psi = \mathrm{id}$. □

J.2 Some facts about operator norm

In this section, we collect some facts about the operator norm that are used in Appendix H.

Lemma 30 (Inversion formula for the operator norm). *Let $M \in \mathbb{R}^{d \times d}$ be an invertible matrix. Then*

$$\left\|M^{-1}\right\|_{\mathrm{op}} = \left(\lambda_{\min}\left(M^\top M\right)\right)^{-1/2} .$$

Proof. By the definition of the operator norm, we know that

$$\left\|M^{-1}\right\|^2_{\mathrm{op}} = \lambda_{\max}\left((M^{-1})^\top M^{-1}\right) .$$

Since we are in a commutative ring, $(M^{-1})^\top = (M^\top)^{-1}$. Additionally, for any two matrices such that AB is invertible, $(AB)^{-1} = B^{-1}A^{-1}$. Therefore

$$\left\|M^{-1}\right\|^2_{\mathrm{op}} = \lambda_{\max}\left((MM^\top)^{-1}\right) .$$

Since $MM^\top$ is a positive definite matrix, $\mathrm{Spec}\left(MM^\top\right) \subseteq \mathbb{R}_+$, and $\lambda_{\max}\left((MM^\top)^{-1}\right) = \lambda_{\min}\left(MM^\top\right)^{-1}$. We can conclude since for any two matrices, AB and BA have the same spectrum. □

Lemma 31 (Bounding the operator norm). *For any matrix $M \in \mathbb{R}^{d \times d}$, we have*

$$\|M\|_{\mathrm{op}} \leq \|M\|_{\mathrm{F}} \leq \sqrt{d}\,\|M\|_{\mathrm{op}} .$$

Proof. We first write

$$\|M\|_{\mathrm{op}} = \lambda_{\max}(M^\top M) \leq \sum_j \lambda_j(M^\top M) = \mathrm{Tr}\left(M^\top M\right) = \|M\|_{\mathrm{F}} .$$

As for the second part of the result, we write

$$\begin{aligned} \|M\|^2_{\mathrm{F}} &= \mathrm{Tr}\left(M^\top M\right) && \text{(definition)} \\ &= \sum_{i=1}^{d} \lambda_i(M^\top M) && \text{(property of the trace)} \\ &\leq d\lambda_{\max}\left(M^\top M\right) && \text{(non-negative eigenvalues)} \end{aligned}$$

□