On the Usage of Gaussian Process for Efficient Data Valuation

Clément Bénesse,[a], Patrick Mesana,[b,c], Athénaïs Gautier,[d], and Sébastien Gambs[b]

[a]Opsci.ai, Paris, France
[b]Université du Québec à Montréal, Montréal, Québec, Canada
[c]HEC Montréal,, Montréal, Québec, Canada
[d]COSMO - Stochastic Mine Planning Laboratory, Department of Mining and Materials Engineering, McGill University, Montreal, Quebec, Canada

Abstract

In machine learning, knowing the impact of a given datum on model training is a fundamental task referred to as Data Valuation. Building on previous works from the literature, we have designed a novel canonical decomposition allowing practitioners to analyze any data valuation method as the combination of two parts: a utility function that captures characteristics from a given model and an aggregation procedure that merges such information. We also propose to use Gaussian Processes as a means to easily access the utility function on "sub-models", which are models trained on a subset of the training set. The strength of our approach stems from both its theoretical grounding in Bayesian theory, and its practical reach, by enabling fast estimation of valuations thanks to efficient update formulae.

1 Introduction

Recently, the field of Machine Learning (ML) has witnessed the development of increasingly complex models and algorithms, due in part to a surge in the amount of available data. However, not all data are created equal, and the quality and relevance of the data play a crucial role in the success of any ML model. Data Valuation (DV) , the process of quantifying the value of data for a specific task, has emerged as a critical concern [Jia et al., 2019b, Ghorbani and Zou, 2019, Ghorbani et al., 2020]. More precisely, the DV field has witnessed

a growing interest in measuring the impact of individual data points on the performance of ML algorithms.

One common approach in this regard is to evaluate how the removal of a specific datum affects the overall model performance. This DV technique, called *Leave One Out* (LOO), provides a measure of the importance of this data point. To capture group effects between inputs, a classical improvement of the LOO is the *Data Shapley* approach [Ghorbani and Zou, 2019]. This method finds its theoretical roots in game theory and considers all *coalitions* or groups of variables, comparing performances with or without the specific datum of interest. However, a critical limitation of this approach lies in its fundamental requirement to compute the performance of a trained ML algorithm for all possible subsets of the training dataset. This requirement, although theoretically informative, often proves to be computationally infeasible due to the exponential growth in the number of subsets as a function of dataset size.

To address this computational challenge, we propose an innovative approach that leverages the strengths of Gaussian Processes (GPs) in the context of DV. GPs [Matheron, 1963, Rasmussen and Williams, 2005] are a versatile tool in ML known for their ability to model complex relationships in data, to provide uncertainty estimates, to seamlessly incorporate new data thanks to analytical update formulae and to make goal-oriented exploration of the input space. In this context, GPs offer a novel solution to the computational bottleneck associated with traditional DV techniques.

In our work, we leverage the unique structure of GPs enables us to perform explicit and immediate computations of sub-models, which are models trained on subsets of the complete dataset. In the framework of DV, this means *that we can easily and at low cost access information on models trained without a given datum of interest, and compare it with the model trained on all data points.* Thus, in balance with the exhaustive approaches required by Data Shapley and similar techniques – even if this can be alleviated by methods such as truncated Data Shapley –, GPs allow us to obtain insights into the impact of individual data points by constructing easily these sub-models. This is achieved through the GPs' distribution inherent ability to be completely summarized by their mean and covariance function, which can both be expressed as explicit weighting of values obtained for the training set. By working within the GPs framework, we circumvent a large part of the computational intractability associated with traditional DV methods, making it feasible to efficiently assess the importance of data points in large datasets. Thus, our approach not only provides practical advantages but also offers a principled and data-driven way to quantify the value of individual data points, contributing to more informed decision-making in data pre-processing, feature selection and model development.

The outline of the paper is as follows. First in Section 2, we review both the DV and GP literature and related work. Then, in Section 3, we propose a canonical decomposition of DV methods as two intertwined elements: a utility function and an aggregation procedure. We then discuss how the Integrated Variance of GPs can be used as a possible utility function to speed up the computations of DV. Finally, in Section 4, we demonstrate experimentally its

usefulness before concluding in Section 5.

2 Background

Notations. We will use the following notations in the rest of the paper. We consider a dataset $\mathcal{D}$ composed of n entries $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n$ that are the realizations of a random variable $\boldsymbol{X}$ that belongs to a probability space $(\mathcal{X}, \mathcal{T}, \mathbb{P}_{\boldsymbol{x}})$. Let $f(\boldsymbol{x}, \mathcal{D})$ denote a model that takes as input a vector $\boldsymbol{x}$ and that is trained using the dataset $\mathcal{D}$. In the ML literature, the model is often obtained by choosing $f(\boldsymbol{x}; \mathcal{D}) = \text{argmin}_{g \in \mathcal{G}} L_n(g, \mathcal{D})$ for $\mathcal{G}$ a certain class of functions and L_n an empirical loss. Considering a set A, let $\mathcal{D}_{\sim A}$ denote the set $\{i \in \mathcal{D}, i \notin A\}$, which is the complementary set of A in the complete dataset – *e.g.* $\mathcal{D}_{\sim i}$ for the complete dataset except for the datum i.

2.1 Related Work on Data Valuation

At its core, Data Valuation (DV) aims at quantifying the contribution of individual data point—or datum—in training a model, assessed using a test set. Originally, the development of DV arises from the increasing desire of individuals and organizations to quantify the value of their data [Jia et al., 2019b], hence the use of the economic term "valuation". While Ghorbani and Zou share this perspective, they also regard DV as a method for assessing data quality [Ghorbani and Zou, 2019]. In particular, DV is useful when training models on noisy datasets, as it helps to identify and remove data points that degrade the model's performance. In addition, as ML gains wider adoption, determining the influence (or lack thereof) of individual data points in data-driven decisions has become crucial. Besides, DV can be instrumental in performing data removal, which can be defined in the ML context as determining the data points that are absolutely necessary to the performance of a model. Doing so can lead to lower storage costs, together with reduced privacy risks.

Shapley values, a concept originated from cooperative game theory [Shapley, 1953], are often employed as a reference and benchmark for distributing value among data. Its strength lies in its fair axioms and the stability of values it provides. In contrast, naive methods like LOO, in which two models are trained – one on the whole dataset and one on the dataset except the datum of interest, have been shown to be less reliable [Ghorbani and Zou, 2019], albeit faster to compute. Indeed, the major drawback of the Shapley values is that calculating them for all data points has an exponential complexity with respect to the size of the dataset, which can render the computation of data values impractical.

Thus, since the introduction of DV in ML, much of the effort has been directed towards making it computationally feasible, juggling between the need to explore numerous subsets to capture datapoint interactions and the cost associated with computations on sub-models. One strategy is to approximate it using Monte Carlo methods [Ghorbani and Zou, 2019]. Another possibility is to adopt alternative concepts such as the Banzhaf [Wang and Jia, 2022] or the

Beta values [Kwon and Zou, 2021]. Their associated aggregation procedures can be linked to the game theory literature on semi-values, of which Shapley values are a special case. However, one of their advantages is that there are more efficient and robust methods for estimating values. Nevertheless, these methods still necessitate retraining the model to compute the contributions of data points. A third approach involves employing less computationally demanding models or specific families of models, such as nearest neighbors algorithms, which due to their structure do not require computing the contribution of data for every subset [Jia et al., 2019a]. In addition, data value can also be defined by its contribution to the distance between training and test datasets, leading to a learning-agnostic DV framework like LAVA, which is computationally more efficient [Just et al., 2023].

2.2 Primer on Gaussian Processes

Originally introduced within the domain of geo-sciences [Krige, 1951, Matheron, 1963], GPs became a popular tool in ML [Rasmussen and Williams, 2005] due to several advantages. First, GPs models are flexible non-parametric models that provide a built-in uncertainty quantification, which can be highly valuable for goal-oriented tasks. Indeed, they enable modeling of a large class of random functions and work on a variety of inputs: continuous, categorical, structured (*e.g.*, molecules, graphs, etc.). As such, they are proxies of choice for performing Bayesian optimization [Snoek et al., 2012], stochastic inversion [Travelletti et al., 2022] and other statistical inference tasks [Marrel et al., 2009]. Traditionally, these models are considered particularly efficient in a low data regime and can provide valuable insights in applications in which data collection is costly or difficult. While originally criticized for their scalability issues, recent improvements in the implementation of GPs have led to a renewal of their use. More precisely, their improved applicability comes from efficient approximation schemes such as Gaussian Markov Random Fields [Lindgren et al., 2011] and variational approaches [Hensman et al., 2013], as well as from improved capacities of the hardware available [Wang et al., 2019] (*i.e.* the use of GPUs rather than CPUs). Hereafter, we review the basics of GP regression (GPR) from a spatial statistics point of view by presenting the linear unbiased estimators and refer the reader to textbooks on the matter such as Rasmussen and Williams [2005] and references therein for the ML-centric perspective.

Basics of GPR. A GP is a collection of random variables, any finite number of which have a joint multivariate Gaussian distribution, which makes it a popular choice as a prior over functions. A GP Z's distribution is characterized by its mean function: $m(\boldsymbol{x}) := \mathbb{E}[Z(\boldsymbol{x})]$ and its covariance kernel $k(\boldsymbol{x}, \boldsymbol{y}) := \mathrm{Cov}(Z(\boldsymbol{x}), Z(\boldsymbol{y}))$, for any $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}$. We represent it by $Z \sim \mathcal{GP}(m, k)$. Hereafter, our focus is on GPR, a non-parametric regression approach. More precisely, we consider a dataset of n pairs of inputs - outputs: $\mathcal{D} = (\boldsymbol{x}_i)_{1 \leq i \leq n}$, along with each observed output $z_i \in \mathbb{R}$ assumed to be a realization of the function to be modeled at $\boldsymbol{x_i}$, with $z_i = \mu(\boldsymbol{x_i}) + Z(\boldsymbol{x_i})$. Here, μ is a trend function modeling the deterministic part of the observation and $Z \sim \mathcal{GP}(0, k)$ is

a centred GP capturing the spatial stochastic dependency. The three primary variants of GPR are: (1) when μ is fully known (*Simple Kriging*), (2) when μ is an unknown constant (*Ordinary Kriging*) or (3) when μ is a linear combination of given basis functions with unknown coefficients (*Universal Kriging*) [Omre and Halvorsen, 1989].

In practical scenarios, the trend is generally unknown, which makes Simple Kriging not suitable. Since Universal Kriging encompasses Ordinary Kriging as a special case, we will focus on it. Assuming that the trend is given by $\mu(\boldsymbol{x}) = \sum_{j=1}^{p} \beta_j f_j(\boldsymbol{x})$, the Universal Kriging predictor at a point $\boldsymbol{x} \in \mathcal{X}$ is linear in the observed values $\boldsymbol{z} = (z_i)_{1 \leq i \leq n}$:

$$\hat{Z}(\boldsymbol{x}) := \boldsymbol{\lambda}(\boldsymbol{x})^\top z, \tag{1}$$

in which the vector of weights $\boldsymbol{\lambda}(\boldsymbol{x})$ is given by:

$$\begin{pmatrix} K & F \\ F^\top & \mathbf{0} \end{pmatrix}^{-1} \begin{pmatrix} \boldsymbol{k}(\boldsymbol{x}) \\ \boldsymbol{f}(\boldsymbol{x}) \end{pmatrix} = \begin{pmatrix} \boldsymbol{\lambda}(\boldsymbol{x}) \\ * \end{pmatrix} \tag{2}$$

with K being the matrix $(k(\boldsymbol{x_i}, \boldsymbol{x_j}))_{1 \leq i,j \leq n}$, F being the matrix $(f_i(\boldsymbol{x_j}))_{1 \leq i \leq p, 1 \leq j \leq n}$, $\boldsymbol{k}(\boldsymbol{x})$ being the vector of $(k(\boldsymbol{x_i}, \boldsymbol{x}))_{1 \leq i \leq n}$ and $\boldsymbol{f}(\boldsymbol{x})$ the vector $(f_i(\boldsymbol{x}))_{1 \leq i \leq p}$. Therefore, the weight for each observation combines insights from $\boldsymbol{f}(\boldsymbol{x})$ pertaining to the trend, as well as from $\boldsymbol{k}(\boldsymbol{x})$ relating to the spatial structure of our model.

Two key features of GP regression are valuable properties for this research. First, the inclusion of a new observation $(\boldsymbol{x_{n+1}}, \boldsymbol{z_{n+1}})$ in $\mathcal{D}$ leads to an updated Kriging weight system, extending the one described in Equation 2. Second, studying the residuals, defined as $Z(\boldsymbol{x}) - \hat{Z}(\boldsymbol{x})$, provides valuable insights on the model's accuracy and effectiveness.

Update formula. The key part of GPR is accessing the inverse of the covariance matrix K. However, when adding new data, this matrix changes and its size increases. Naively, one would need to invert the new matrix that includes the new points, without any prior information. This method can be quite expensive, especially when adding few points to an otherwise large dataset – *e.g.* adding one point to a dataset of 1000 points would lead to the inversion of a 1001×1001 matrix regardless of the knowledge on the covariance sub-matrix of size 1000×1000. This issue can be addressed using the Schur complement, leveraging already known information on the covariance matrix to update it efficiently when adding datapoints. This result can be found in Ginsbourger and Schärer [2023].

Insights on the residuals. In our case, we are also interested in the Integrated Variance of the GP predictor. The residuals' variance $Var(Z(\boldsymbol{x}) - \hat{Z}_{\mathcal{D}}(\boldsymbol{x}))$ quantifies the uncertainty of the predictor $\hat{Z}_{\mathcal{D}}$ in its prediction at a given point $\boldsymbol{x}$. As more and more datapoints are used for the predictor, predictions become closer to realizations of the model and the quantity $Var(Z(\boldsymbol{x}) - \hat{Z}_{\mathcal{D}}(\boldsymbol{x}))$ decreases for each $\boldsymbol{x}$. This quantity is directly accessible by computing the

inverse of the matrix

$$M_{\mathcal{D}}(\boldsymbol{x}) = \begin{pmatrix} k(\boldsymbol{x},\boldsymbol{x}) & k(\boldsymbol{x},\boldsymbol{x}_{\mathcal{D}}) & F(\boldsymbol{x}) \\ \boldsymbol{k}(x) & K & F \\ F(\boldsymbol{x})^T & F & 0 \end{pmatrix}. \tag{3}$$

Note that, again, we use the covariance matrix K whose inverse is readily accessible and whose update can be done following the previous point. We are also interested in the global uncertainty of the predictor. This is quantified by the Integrated Variance, which can be understood as the aggregated remaining uncertainty of the predictor:

$$IV(\mathcal{D}) := \int Var(Z - \hat{Z}_{\mathcal{D}}) d\mathbb{P}_{\mathbb{X}}. \tag{4}$$

This can be expressed in terms of coefficients of $M_{\mathcal{D}}^{-1}(\boldsymbol{x})$, by denoting $M_{D}^{-1}(\boldsymbol{x})[1,1]$ the top left coefficient and taking:

$$IV(\mathcal{D}) = \int_{\boldsymbol{x}\in\mathcal{X}} M_{\mathcal{D}}^{-1}(\boldsymbol{x})[1,1] d\mathbb{P}_{\mathbb{X}}(\boldsymbol{x}). \tag{5}$$

However, inverting the matrix $M_D(\boldsymbol{x})$ for any given input point $\boldsymbol{x}$ can be costly. This issue can be, once again, solved by using the Schur complement of the covariance matrix.

Theorem 2.1 (Iterative update of Integrated Variance)**.** *For a set $A \subset \mathcal{D}$, let $\tilde{K}_A := M_A[\sim 1, \sim 1]$ the matrix obtained by deletion of first row and column of M_A and the covariance matrix of the data point included in A. When adding a datum i, the inverse of the matrix $\tilde{K}_{A\cup\{i\}}$ can be obtained using the Schur complement formula knowing only the inverse of K_A – see Ginsbourger and Schärer [2023] or explicit formula in Appendix A. Moreover, let $IV(A)$ be the integrated variance of the GP from the dataset $A \subset \mathcal{D}$.Then we have an explicit formula for $IV(A \cup \{i\})$ that we provide in Equation 13 in Appendix A.*

More precisely, one can see two different uses of the Schur complement here: one to update the covariance matrix and one to extract the quantity $Var(Z - \hat{Z}_{\mathcal{D}})$.

3 Proposed Data Valuation Framework

In this section, we describe our proposed framework for showing that most, if not all, DV tools can be decomposed in two parts that we describe in the following subsection. This dichotomy allows to conduct a generic and systemic analysis of DV tools while providing the practitioner with additional insights and knowledge on what is actually captured by these indices. Then, we provide several examples of different DV methods, comparing the differences between them, along with some hints on classical limitations that they have when confronted with real world datasets (*e.g.*, the need for sub-models).

3.1 Utility Functions and Aggregations

As discussed previously, there are various DV indices that have been proposed in the literature, but as we detail hereafter the vast majority of them can be decomposed as two elements.

A utility function. This corresponds to a way of measuring some characteristic of a model, which is usually done through a function Φ quantifying the model performance. Common choices for this performance measure include the accuracy, the RMSE, the F1-score, the area under the ROC curve (AUC), etc. However, some indices can be simpler (*e.g.*, the model output at a given point can be used directly) or more complex (*e.g.*, a composite metric of the fairness of the model). Similarly, this utility function can be either computed using a test set – for instance, after an initial split of the data done before the DV procedure (*e.g.*, the accuracy of the model with respect to the test set) – or using exact computation of characteristics (*e.g.*, the expectation of a given functional of the model such as a fairness criterion). We emphasize that in this canonical decomposition, a usual train-test split of the data as commonly used in the literature is done before this procedure, with train data (*i.e.*, $\mathcal{D}$ used for model training for obtaining $f(\boldsymbol{x}; \mathcal{D})$) – and test data used for computing later quantities – such as $\Phi[f(\cdot; \mathcal{D})]$ if such a utility function needs empirical data. The choice of the utility function is the most flexible element of DV and usually the part in which practitioners can use expert knowledge the most easily to choose this function.

Note that in some situations, this utility function is only computed on some specific subsets of the complete dataset $\mathcal{D}$ according to the needs of the aggregation part (which we discuss afterwards) (*e.g.*, $\Phi[f(\boldsymbol{x}; \mathcal{D})]$ and $\Phi[f(\boldsymbol{x}; \mathcal{D}_{\sim i})]$ for any datum i). However, in general settings, this quantity needs to be computed for all models trained from subsets of the dataset of $\mathcal{D}$. More precisely, generally we need to be able to compute $\Phi[f(\boldsymbol{x}; A)]$ for any $A \subset \mathcal{D}$, which can be expensive and is our motivation for GP usage.

An aggregation procedure. Once we have the value of the utility function on all possible subsets of $\mathcal{D}$, we need to compare them depending on the presence or absence of the datum i to identify its influence on the training of the model. The simplest of these aggregation procedures is the LOO described previously, in the sense that the valuation of the datum is the direct comparison between performances of these two models. However, by doing this, we are limited to direct effects of the datum's presence in the dataset. Various refinements have been proposed to take into account indirect effects – that is belonging of the datum to specific subsets. Among them, one can think about Shapley values [Shapley, 1953] and recent variations such as Banzhaf [Wang and Jia, 2022] or Beta Shapley values [Kwon and Zou, 2021] found in DV literature. Nonetheless, other aggregation values can be of interest, including more general semi-values and other concepts from cooperative game theory. Note that the aggregation procedure indicates how many sub-models are needed for accessing DV indices. For instance, simple aggregation procedures may not be very informative or may only provide first-order insights but can be computed with very

few sub-models.

To formalize this framework, we first introduce the following definition.

Definition 3.1 (General data valuation of a set of indices)**.** Let $\Phi : \mathbb{L}^2(\mathbb{P}_{\boldsymbol{X}}) \mapsto \mathbb{R}$ be a utility function and $Agg_i : \mathcal{P}(\mathcal{D}) \mapsto \mathbb{R}$ be an aggregation procedure for the datum i. Then a *general Data Valuation set of indices* $\{DV_i(\Phi, Agg_i, f, \mathcal{D})\}$ is defined by its elements

$$DV_i(\Phi, Agg_i, f, \mathcal{D}) := Agg_i(\{\Phi(f(\boldsymbol{x}; A)), A \subset \mathcal{D}\}). \tag{6}$$

Note that the function Agg_i depends on the datum of interest but can be applied to all the quantities $\Phi[f(\boldsymbol{x}; A)], A \subset \mathcal{D}$. Figure 1 also illustrates the envisioned pipeline with an example in which the aggregation procedure includes all coalitions to which the datum i belongs.

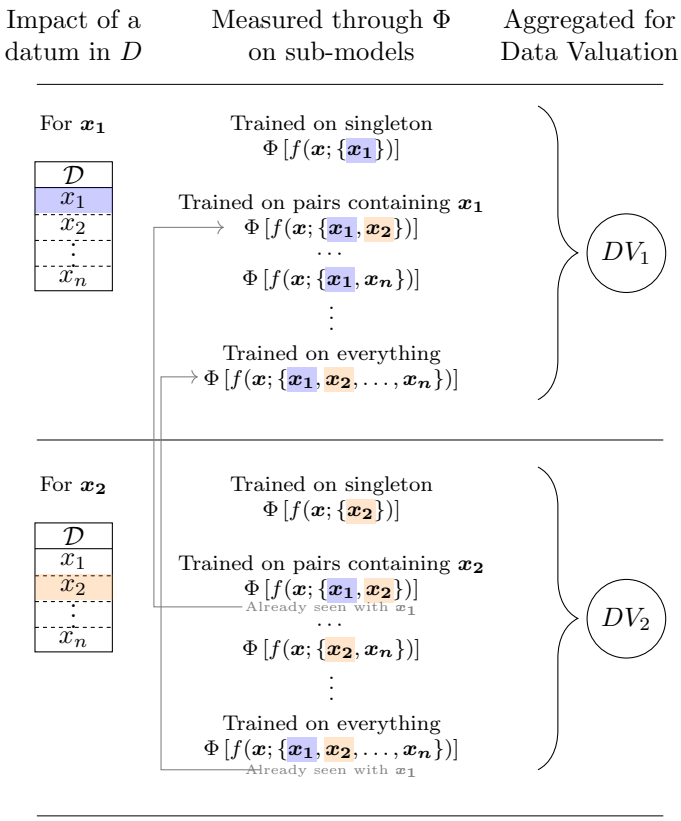


Figure 1: Generic pipeline for Data Valuation.

The main difficulty for performing DV is to access the quantity $\Phi[f(\boldsymbol{x}); A]$ for some subset A of $\mathcal{D}$ when working with meaningful utility functions. In the general case, the training is not done for all subsets of the data and the

intermediary models and values are not kept. However, in this paper we chose to focus on the GPs as a means to alleviate this problem.

3.2 Examples of DV indices

Hereafter, we describe some examples of DV indices that can be used in practice. Following our proposed framework, these indices can be interpreted as a specific choice of utility function and aggregation procedure, thus allowing them to capture specific characteristics of the dataset.

Naive utility function, naive aggregation. The first example is the classical DV procedure that arises in the classification framework. In that case, we have a random variable Y that is predicted from covariates $\mathbf{X}$ thanks to the model f. The utility function in the DV procedure is the accuracy of the trained model and the aggregation procedure simply computes the difference between the utility function on the model trained with the complete dataset and the utility function taken on the model trained with the truncated dataset without the datum of interest. More precisely, we have:

$$\Phi[f(\boldsymbol{x}; A)] = \int \mathbb{1}_{f(\boldsymbol{x};A)=Y} d\mathbb{P}_{\boldsymbol{x}}(\boldsymbol{x}); \tag{7}$$

$$Agg_i(\{\Phi[f(\boldsymbol{x}; A)]; A \subset \mathcal{D}\}) = \Phi[f(\boldsymbol{x}; \mathcal{D}] - \Phi[f(\boldsymbol{x}; \mathcal{D}_{\sim i}]. \tag{8}$$

In terms of benefits, this set DV indices is simpler to interpret and compare than more complex set of indices. It also needs only $n + 1$ model training to obtain all valuations, namely the utility function on the complete dataset and n utility functions on truncated datasets. However, one of the biggest drawbacks of this technique is that it quantifies only "first-order" importance of a datum – which means that the datum only provides the information of its contribution when added to an almost complete dataset. Other more subtle effect – *e.g.* when a datum is interesting for the complementary information it provides when linked with specific subsets of data – cannot be captured by such indices. Additionally, another issue is that the valuations obtained do not directly sum to one and have no reason to be positive as additional data may deteriorate the accuracy of the model.

Data Shapley Data Shapley indices [Ghorbani and Zou, 2019] are one of the most used DV techniques in the literature. The utility function for this set of indices can be readily changed according to the use case and is not the core of this valuation. However, the aggregation procedure originated from game theory and can be linked with several other fields such as explainability. Here, the objective is to go beyond the "first-order" aggregation explained in the previous example by taking into account the importance of the datum of interest not only in the complete dataset but also in every possible subsets (or coalitions) of this dataset. To match a certain number of axioms defined by Shapley [1953], the weight of the utility functions are uniquely defined and allow to obtain a set of data valuations summing to one. Additionally, the valuations obtained can be easily interpreted as the coalitional importance of each datum in the training

dataset. The aggregation procedure is given by the following formula:

$$DS_i(\Phi, f, \mathcal{D}) := (n!)^{-1} \sum_{S \in \mathcal{D}_{\sim i}} \Phi[f(\boldsymbol{x}; S \cup \{i\})] - \Phi[f(\boldsymbol{x}; S]. \tag{9}$$

Note that this definition is equivalent to other definitions that focus on coalitions enumeration – given for instance in the literature [Ghorbani et al., 2020] – by considering all possible permutations, a classical alternative vision for Shapley values. Thanks to this formulation, it allows for an easier understanding of the Monte-Carlo estimator of this aggregation procedure as this estimator is a direct empirical plug-in in this equation. The most problematic part of these semi-values based indices is the need to evaluate the utility function on all possible coalitions – that is to say, on all 2^n possible trained models. As this is usually computationally unfeasible, several estimations have been proposed to alleviate this issue [Ghorbani et al., 2020]. Later, we will use the Monte Carlo estimator found in particular in Ghorbani and Zou [2019] for DV or Da Veiga et al. [2021] in Explainability.

Integrated variance and Gaussian processes.

The final example introduced here is the novelty of our approach. Its aim is to give a utility function to the practitioner – more precisely the Integrated Variance of the Gaussian Process – that is readily understandable. Such a quantity is rooted in Bayesian theory as a measure of uncertainty of the trained model and its training dataset, compared with the true phenomenon modeled. While the aggregation procedure could easily be chosen as the naive one, we consider the more challenging and informative aggregation function of the DataShapley indices. One of the main issue, as evoked above, is the high number of trained models needed. While this is still true when using Gaussian Processes, we can obtain analytical expressions for the Integrated Variance and its updates when adding a datum. This is what makes the computation of the utility function faster, since we do not have to compute everything from scratch each time we consider a new coalition. Furthermore by clever usage of the Schur complement – detailed in Appendix A, see Zhang [2006], Gallier [2010] for in-depth details –, we propose an algorithm to obtain results that are computationally realistic and allow for better estimation due to the possibility of more calls to the model in the Monte Carlo loop of the estimator, since we do not spend as much compute for the utility function. We emphasize that this framework does not speed-up the aggregation procedure but rather accelerate the computation of the utility function, allowing to spend a greater computational budget in the aggregation procedure (*e.g.*, more runs in the Monte-Carlo loop).

Note that a final benefit of using GPs is that it is versatile while proposing a unified framework for working with tabular and non-tabular data. Indeed, depending on the type of data, the only part that needs to be adapted for the GP to work is the kernel. This subject has been studied in the literature in recent years, especially when considering GP modeling in biochemistry [Tanimoto, 1958, Tripp et al., 2023]. This can be interesting for broader applications of DV, but we leave it as future works.

Algorithm 1: Integrated Variance and Shapley Aggregation estimator

Data: Training set $\mathcal{D}$, full kernel matrix $M_{\mathcal{D}}$, a maximal budget b_{max}, a tolerance threshold ε

Result: Data valuation of training points $DV = \{DV_i, i \in \{1, \cdots, n\}\}$

Initialize $DV_i \leftarrow 0$ for all $i \in \{1, \cdots, n\}$;

Initialize $b \leftarrow 0$;

while $b \leq b_{max}$ **do**

  Initialize $\text{Upd}_i \leftarrow 0$ for all $\{i \in \{1, \cdots, n\}\}$;

  Initialize $\text{IV}_{\pi,\emptyset} \leftarrow IV(\emptyset)$;

  Update the budget: $b \leftarrow b + 1$;

  Draw a random permutation π of data points;

  for $j \in \{1, \cdots, n\}$ **do**

    Use $\text{IV}_{\pi,j-1} := IV(\{\pi[1], \cdots, \pi[j-1]\})$ to compute $\text{IV}_{\pi,j} := IV(\{\pi[1], \cdots, \pi[j]\})$ with Equation 13 from Appendix A;

    Update $\text{Upd}_j \leftarrow \frac{1}{n!}(\text{IV}_{\pi,j} - \text{IV}_{\pi,j-1})$;

  end

  Update $DV_i \leftarrow \frac{b-1}{b} DV_i + \frac{1}{b} \text{Upd}_i$ for all $\{i \in \{1, \cdots, n\}\}$;

end

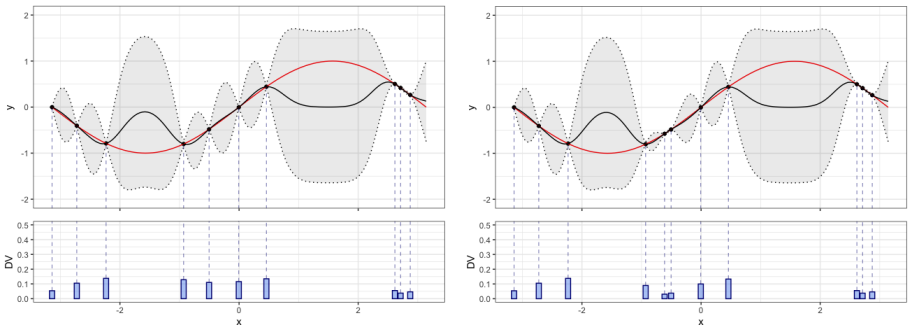
4 Experiments

In our exploration of DV using GPs, we focus on synthetic data and the Boston Housing dataset [Harrison and Rubinfeld, 1978], a widely recognized benchmark in regression tasks. We also refer the reader to Appendix B for additional experiments on Boston Housing and for another regression task performed on a variant of the Adult Income Dataset generated through the Folktables framework [Ding et al., 2021].

4.1 Synthetic data

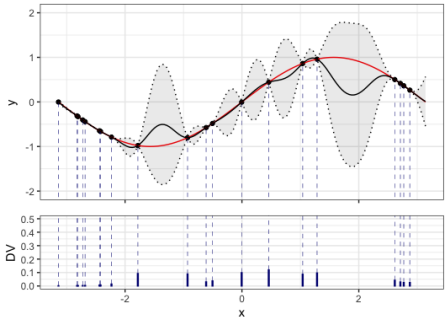
We consider the scenario in which data points have been used to train a model f represented in Figure 2 as the red curve – here, a sinus function. Following the procedure described in the previous section, we use a GP with a Gaussian kernel – $k(x, x') \propto \exp(-(x - x')^2/2)$. We perform GPR and at each step, we leverage Schur complement to integrate new datapoints.We compute the DV with IV as a utility function. The results are showcased in Figure 2, where we visualise the evolution of the GPR predictor, the uncertainty reduction when adding new points, and the DV indices.

As expected, we can observe that the points with the biggest importance are isolated points. This is due to the fact that removal of an isolated point significantly increases the uncertainty of the neighbouring area. Conversely, we can see that point clusters usually contains individuals with low valuation since their contribution is similar one to another. More precisely as a whole, the



(a) GP with 10 datapoints.

(b) GP with 11 datapoints.



(c) GP with 20 datapoints.

Figure 2: Top panels represent the data assimilation of a Gaussian Process. The true model is in red, the predictor in black and the point-wise 95% confidence intervals are represented by the grey ribbon. Bottom panels plot the DV of each datum, with the procedure described in subsection 4.1.

cluster can be highly influential but this influence will be split between several individuals.

4.2 Boston Housing

The Boston Housing dataset consists of 506 records with 14 features each, with the objective of predicting the median value of homes in Boston suburbs. This dataset was chosen for its relevance to regression tasks and its comprehensive range of features, such as crime rate and property tax rate. We perform GPR on noisy observations with a Matérn kernel, reflecting common practices in the literature.The conducted experiments are Leave-One-Out (LOO), LOO with Schur complement, Data Shapley Value computed without Schur (DSV) and DSV with Schur complement, aiming to evaluate individual data contributions and improve computational efficiency. The Schur complement's integration is pivotal in managing the covariance matrix's dynamic nature, especially during DSV calculations, balancing computational stability with efficiency (Appendix B).

Our analysis for the Boston Housing dataset highlights the importance of using the Schur complement, thus maintaining efficiency without compromising valuation integrity. The sanity of our approaches is demonstrated by the consistent results throughout all methods. Our implementation yields the same rankings and LOO values no matter the approach considered, as confirmed by a perfect Spearman's coefficient. Additionally, a data removal experiment demonstrates that DSV-based strategies significantly outperformrandom data removal, especially when exceeding 50% dataset reduction, highlighting the effectiveness of data valuation in scenarii of substantial data limitation (*cf.*, Figure 3). The appendix reveals intricate details concerning the parametrization of the kernel, and computational strategies. In particular, we show that varying kernel noise levels can be used for smoothing the model's output, thus controlling the model's sensitivity to individual data points. We also discuss the role of a covariance reset strategy in enhancing the computational stability of DSV valuations with Schur complement (Appendix B.1). Finally, in our current benchmark we notice disparities between Data Shapley values when using Schur complement, highlighting that strategic adjustments in the calculation can lead to significant variations in data valuation.

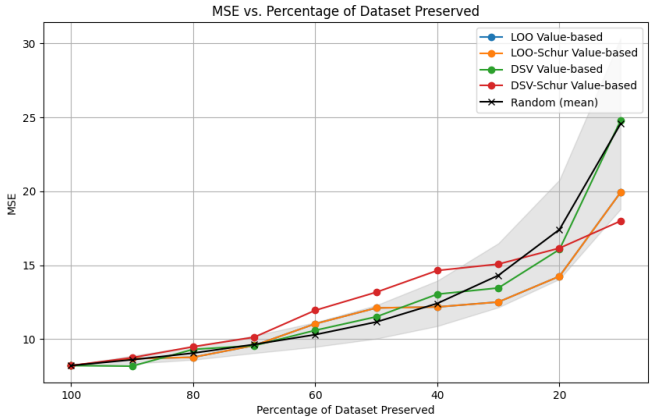


Figure 3: Comparison of MSE as a function of the percentage of the Boston Housing dataset preserved. The graph illustrates the performance of data valuation methods—LOO, LOO-Schur, DSV, and DSV-Schur against the baseline of random data point removal (mean). Note that LOO and LOO-Schur are equal. The MSE increases for all methods as more data is removed, but the valuation-based methods consistently outperform random removal, with a notable advantage as less of 20% is retained.

5 Conclusion and Future Works

Our contribution in this paper is twofold. First, we have introduced a unifying framework of DV through a canonical decomposition of DV as the combination of two elements: a utility function that extracts some informative characteristics from a model and an aggregation procedure that merges information from values taken by the utility function on all models trained on a subset of the dataset. Second, we tackled the issue of combinatorial explosion when accessing such models trained on subsets, and have proposed to use of GPs as a means to reduce costs by having readily tractable "sub-models". Our careful handling of the algebra in GPR yields low cost estimations of the data valuations. We focused on one setting where these two axis synergize, by detailing the special case of Integrated Variance as a utility function combined with the Shapley aggregation procedure. We demonstrated the applicability of this approach on synthetic data as well as two others datasets.

These results pave the road for further developments. First, we believe that our canonical decomposition may be a key component for further systematic analysis of DV indices. Second, update formulae are a versatile tool that can be used to measure the influence of a coalition, in addition to that of a single datum. Finally, GPs usage open a novel and extremely flexible framework for DV, and allows for working with non-tabular data, thus enabling DV in complex dataset.

References

Sébastien Da Veiga, Fabrice Gamboa, Bertrand Iooss, and Clémentine Prieur. *Basics and trends in sensitivity analysis: Theory and practice in R*. SIAM, ., 2021.

Frances Ding, Moritz Hardt, John Miller, and Ludwig Schmidt. Retiring adult: New datasets for fair machine learning. *Advances in Neural Information Processing Systems*, 34, 2021.

Jean H Gallier. Notes on the schur complement. *University of Pennsylvania*, 2010.

Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2242–2251, ., 09–15 Jun 2019. PMLR. URL `https://proceedings.mlr.press/v97/ghorbani19c.html`.

Amirata Ghorbani, Michael Kim, and James Zou. A distributional framework for data valuation. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3535–3544, ., 13–18 Jul

2020. PMLR. URL https://proceedings.mlr.press/v119/ghorbani20a.html.

David Ginsbourger and Cedric Schärer. Fast calculation of gaussian process multiple-fold cross-validation residuals and their covariances, 2023.

David Harrison and Daniel L Rubinfeld. Hedonic housing prices and the demand for clean air. *Journal of Environmental Economics and Management*, 5(1):81–102, March 1978. ISSN 0095-0696. doi: 10.1016/0095-0696(78)90006-2. URL https://www.sciencedirect.com/science/article/pii/0095069678900062.

James Hensman, Nicolò Fusi, and Neil D. Lawrence. Gaussian processes for big data. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, UAI'13, page 282–290, Arlington, Virginia, USA, 2013. AUAI Press.

Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nezihe Merve Gurel, Bo Li, Ce Zhang, Costas J Spanos, and Dawn Song. Efficient task-specific data valuation for nearest neighbor algorithms. *arXiv preprint arXiv:1908.08619*, 2019a.

Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nick Hynes, Nezihe Merve Gürel, Bo Li, Ce Zhang, Dawn Song, and Costas J Spanos. Towards efficient data valuation based on the shapley value. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1167–1176. PMLR, 2019b.

Hoang Anh Just, Feiyang Kang, Jiachen T Wang, Yi Zeng, Myeongseob Ko, Ming Jin, and Ruoxi Jia. Lava: Data valuation without pre-specified learning algorithms. *arXiv preprint arXiv:2305.00054*, 2023.

Daniel G Krige. A statistical approach to some basic mine valuation problems on the witwatersrand. *Journal of the Southern African Institute of Mining and Metallurgy*, 52(6):119–139, 1951.

Yongchan Kwon and James Zou. Beta shapley: a unified and noise-reduced data valuation framework for machine learning. *arXiv preprint arXiv:2110.14049*, 2021.

Finn Lindgren, Håvard Rue, and Johan Lindström. An explicit link between gaussian fields and gaussian markov random fields: the stochastic partial differential equation approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(4):423–498, 2011. doi: https://doi.org/10.1111/j.1467-9868.2011.00777.x. URL https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9868.2011.00777.x.

Amandine Marrel, Bertrand Iooss, Béatrice Laurent, and Olivier Roustant. Calculations of Sobol indices for the Gaussian process metamodel. *Reliability*

Engineering & System Safety, 94(3):742–751, March 2009. ISSN 0951-8320. doi: 10.1016/j.ress.2008.07.008.

Georges Matheron. Principles of geostatistics. *Economic geology*, 58(8):1246–1266, 1963.

Henning Omre and Kjetil B Halvorsen. The bayesian bridge between simple and universal kriging. *Mathematical Geology*, 21:767–786, 1989.

C.E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning series. MIT Press, 2005. ISBN 9780262182539. URL https://books.google.ca/books?id=H3aMEAAAQBAJ.

L. S. Shapley. *17. A Value for n-Person Games*, pages 307–318. Princeton University Press, Princeton, 1953. ISBN 9781400881970. URL https://doi.org/10.1515/9781400881970-018.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper_files/paper/2012/file/05311655a15b75fab86956663e1819cd-Paper.pdf.

Taffee T Tanimoto. Elementary mathematical theory of classification and prediction. Technical report, International Business Machines Corp., 1958.

Cédric Travelletti, David Ginsbourger, and Niklas Linde. Uncertainty Quantification and Experimental Design for Large-Scale Linear Inverse Problems under Gaussian Process Priors, August 2022. URL http://arxiv.org/abs/2109.03457. arXiv:2109.03457 [cs, stat].

Austin Tripp, Sergio Bacallado, Sukriti Singh, and José Miguel Hernández-Lobato. Tanimoto random features for scalable molecular machine learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, ., 2023. NeurIPS. URL https://openreview.net/forum?id=MVOINFAKGq.

Ke Wang, Geoff Pleiss, Jacob Gardner, Stephen Tyree, Kilian Q Weinberger, and Andrew Gordon Wilson. Exact gaussian processes on a million data points. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/01ce84968c6969bdd5d51c5eeaa3946a-Paper.pdf.

Tianhao Wang and Ruoxi Jia. Data banzhaf: A data valuation framework with maximal robustness to learning stochasticity. *arXiv preprint arXiv:2205.15466*, 2022.

Fuzhen Zhang. *The Schur complement and its applications*, volume 4. Springer Science & Business Media, 2006.

A Schur decomposition

In this appendix, we provide more details on the Schur complement, along with the proof of Theorem 2.1.

First, the Schur complement states the following: for a given invertible matrix we have

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix}^{-1} = \begin{pmatrix} (A - BD^{-1}C)^{-1} & -(A - BD^{-1}C)^{-1}BD - 1 \\ -D^{-1}C(A - BD^{-1}C)^{-1} & D^{-1} - D^{-1}C(A - BD^{-1}C)^{-1}BD^{-1} \end{pmatrix}. \tag{10}$$

In the context of the covariance matrix of a Gaussian Process, this means that when adding a point i to a set A, we have:

$$K_{A\cup\{i\}}^{-1} = \begin{pmatrix} k(x_i, x_i) & k(x_i, x_A) & F(x_i)^T \\ k(x_A, x_i) & k(x_A, x_A) & F(x_A)^T \\ F(x_i) & F(x_A) & 0 \end{pmatrix}^{-1} = \begin{pmatrix} \Delta_1 & \Delta_2 \\ \Delta_3 & \Delta_4 \end{pmatrix} \tag{11}$$

with

$$\begin{aligned}
\Delta_1 &= \left(k(x_i, x_i) - \left(k(x_i, x_A)F(x_i)^T\right) K_A^{-1} \left(k(x_A, x_i)F(x_i)\right)^T\right)^{-1}, \\
\Delta_2 &= -(k(x_i, x_i) - \left(k(x_i, x_A)F(x_i)^T\right) K_A^{-1} \left(k(x_A, x_i)F(x_i)\right)^T)^{-1} \left(k(x_i, x_A)F(x_i)^T\right) K_A^{-1}, \\
\Delta_3 &= -K_A^{-1} \left(k(x_A, x_i)F(x_i)\right)^T \left(k(x_i, x_i) - \left(k(x_i, x_A)F(x_i)^T\right) K_A^{-1} \left(k(x_A, x_i)F(x_i)\right)^T\right)^{-1}, \\
\Delta_4 &= K_A^{-1} - K_A^{-1} \left(k(x_A, x_i)F(x_i)\right)^T \left(k(x_i, x_i) - \left(k(x_i, x_A)F(x_i)^T\right) K_A^{-1} \left(k(x_A, x_i)F(x_i)\right)^T\right)^{-1} \\
&\qquad \left(k(x_i, x_A)F(x_i)^T\right) K_A^{-1}.
\end{aligned} \tag{12}$$

Note that this only inverse in these formulae is the inverse of the lesser covariance matrix denoted as K_A^{-1}, which is known from the previous step in our iteration.

Now, when working with the Integrated Variance, we are interested in the top-left term of the inverse of the matrix $M_A(x)$. Using the Schur decomposition on the matrix $M_A(x)$, one readily obtains that

$$M_{A\cup i}(x)^{-1}[1,1] = \left(k(x,x) - \left(k(x, x_i)k(x, x_A)F(x)^T\right) K_{A\cup i}^{-1} \left(k(x_i, x)k(x_A, x)F(x)^T\right)^T\right)^{-1}. \tag{13}$$

This quantity is readily available from the previous development and can be further expanded in lesser terms by developing the products.

B Experiments

This appendix provides additional details regarding the experimental setup.

B.1 Boston Housing

The Truncated DSV algorithm was configured to perform 1500 iterations for the Boston Housing dataset, with a burn-in period of 50 indices per iteration. The burn-in parameter allows the DSV computation to stabilize, aligning closer to Leave-One-Out (LOO) results.

We observed that incorporating noise into the kernel contributes to computational stability at the cost of confidence, as evidenced by a wider Integrated Variance (IV) range when we remove points as shown in Figure 5. The influence of individual data points on IV becomes more discernible with increased noise levels in the kernel. The computational stability was studied by observing the condition number of the covariance matrix (*cf.*, Figure 6). It is important to note that adding noise to the kernel allows to control stability but also affects utility.

We also implemented a covariance resetting strategy for instances where the absolute contribution to IV was too large because of computational instability. This technique allowed the computations using the Schur complement to align more closely with the DSV values.

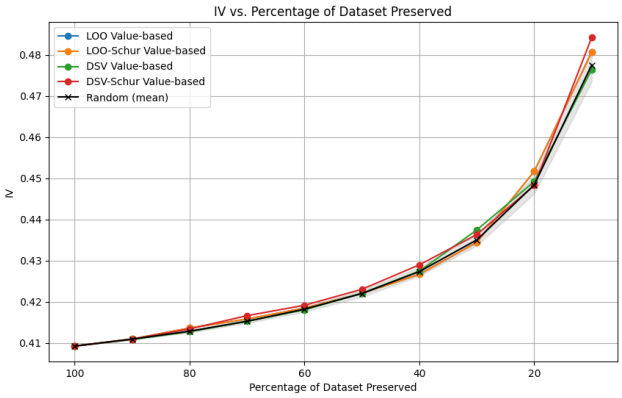


Figure 4: Data valuation impact on Integrated Variance with a kernel noise level of 0.01.

B.2 Folktables

The Folktables dataset, a variant of the Adult Income dataset, was used for predicting the numerical income values of individuals. For this dataset, the Truncated DSV algorithm was configured to perform only 10 iterations because of the dataset size (2000 data points).

Regarding the Folktables results, we noted that while LOO values were closely aligned, DSV exhibited disparities (Low Spearman coefficient) when the Schur complement was utilized. This indicates that the numerical instability,

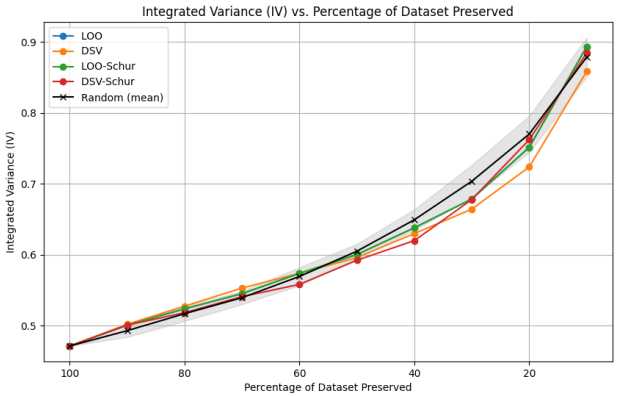


Figure 5: Data valuation impact on Integrated Variance with a kernel noise level of 0.38.

along with the choice of kernel and DSV parameters, can significantly influence empirical outcomes.

The data removal task on the Folktables did not conclusively show that removal based on value-based strategies had an advantage as shown in Figure 7, which suggests that kernel selection and DSV parameterization are critical to the success of these methods.

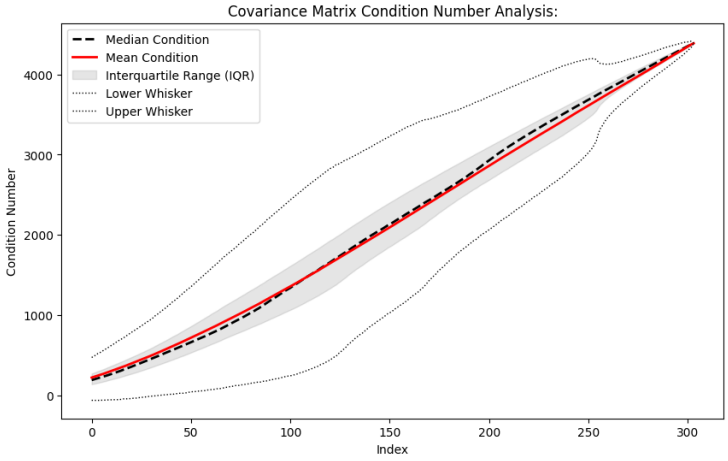


Figure 6: Computational stability analysis using the condition number of the covariance matrix.

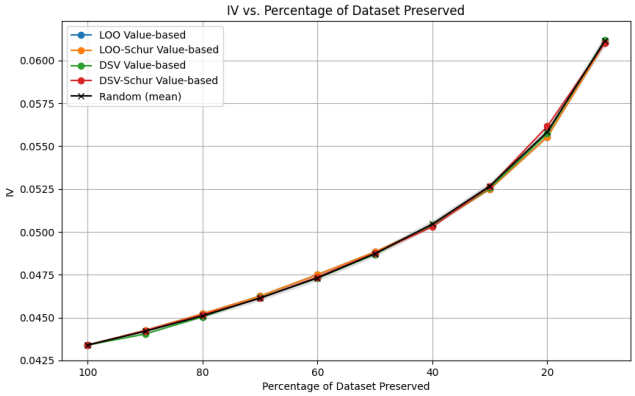


Figure 7: Data removal task results on the Folktables dataset, showing IV as a function of the dataset percentage preserved.