Skoltech
Skolkovo Institute of Science and Technology
Skolkovo Institute of Science and Technology

MULTI-FIDELITY CLASSIFICATION AND ACTIVE SEARCH

Doctoral Thesis

by
NIKITA KLYUCHNIKOV

DOCTORAL PROGRAM IN
COMPUTATIONAL AND DATA SCIENCE
AND ENGINEERING

Supervisor
Associate Professor Evgeny Burnaev

Moscow - 2021

I hereby declare that the work presented in this thesis was carried out by myself at Skolkovo Institute of Science and Technology, Moscow, except where due acknowledgement is made, and has not been submitted for any other degree.

Candidate (Nikita Klyuchnikov)

Supervisor (Prof. Evgeny Burnaev)

Abstract

My thesis is devoted to two special cases of machine learning problems, namely classification and active search, when multiple sources of data with variable fidelity and sampling costs are available. For the multi-fidelity classification problem, I propose to model probabilities of classes with Gaussian processes and combine different data sources by applying a co-kriging schema on them. Since the model is analytically intractable, I derive a fast numeric inference method based on Laplace approximation for this model. For the multi-fidelity active search problem, I propose an algorithm based on Upper Confidence Bound acquisition criterion and co-kriging model that guides the search process by jointly exploring and exploiting low- and high- fidelity sources. The proposed models are evaluated in a series of numerical experiments, that include sensitivity to hyper-parameters, and comparison with baselines and state-of-the-art alternatives on various datasets. Finally, the methodology and examples of applying multi-fidelity classification and active search methods are demonstrated in several industrial data-driven projects.

Publications

1. Klyuchnikov, N., Mottin, D., Koutrika, G., Müller, E. and Karras, P., 2019, July. Figuring out the User in a Few Steps: Bayesian Multifidelity Active Search with Cokriging. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 686-695.

2. Klyuchnikov, N. and Burnaev, E., 2020. Gaussian process classification for variable fidelity data. Neurocomputing, 397, pp.345-355.

3. Klyuchnikov, N., Zaytsev, A., Gruzdev, A., Ovchinnikov, G., Antipova, K., Ismailova, L., Muravleva, E., Burnaev, E., Semenikhin, A., Cherepanov, A. and Koryabkin, V., 2019. Data-driven model for the identification of the rock type at a drilling bit. Journal of Petroleum Science and Engineering, 178, pp.506-516.

4. Gurina, E., Klyuchnikov, N., Zaytsev, A., Romanenkova, E., Antipova, K., Simon, I., Makarov, V. and Koroteev, D., 2020. Application of machine learning to accidents detection at directional drilling. Journal of Petroleum Science and Engineering, 184, p.106519.

5. Romanenkova, E., Zaytsev, A., Klyuchnikov, N., Gruzdev, A., Antipova, K., Ismailova, L., Burnaev, E., Semenikhin, A., Koryabkin, V., Simon, I. and Koroteev, D., 2020. Real-Time Data-Driven Detection of the Rock-Type Alteration During a Directional Drilling. IEEE Geoscience and Remote Sensing Letters, vol. 17, no. 11, pp. 1861-1865.

6. Antipova, K., Klyuchnikov, N., Zaytsev, A., Gurina, E., Romanenkova, E. and Koroteev, D., 2019, September. Data-Driven Model for the Drilling Accidents Prediction. In SPE Annual Technical Conference and Exhibition. Society of Petroleum Engineers.

7. Baranov, A., Burnaev, E., Derkach, D., Filatov, A., Klyuchnikov, N., Lantwin, O., Ratnikov, F., Ustyuzhanin, A. and Zaitsev, A., 2017, December. Optimising the active muon shield for the SHiP experiment at CERN. In Journal of Physics: Conference Series, vol. 934, no. 1, p. 012050. IOP Publishing.

Acknowledgements

I am very grateful to my supervisor, Evgeny Burnaev, for the constant support during the PhD studies and for being actively concerned about my research progress.

I would also like to express appreciation of the constructive feedback that was provided by the reviewers: Andrzej Cichocki, Dmitry Yarotsky, Ivan Oseledets, Maurizio Filippone, Maxim Panov, and Yarin Gal.

Next, I am thankful to Dmitry Koroteev and Alexey Zaytsev for involving me into a series of industrial data-driven projects, in which the developed multi-fidelity methods evinced their utility.

I would also like to thank Panagiotis Karras for initiating my PhD studies at Skoltech and for facilitating my international scientific collaboration.

Next, I would like to acknowledge Ivan Nazarov for useful Python code tips.

I am also very much obliged to all collaborators, who contributed to the projects described in this thesis.

Finally, I am thankful to Skoltech for organizing and providing an environment for conducting state-of-the-art research.

Contents

List of Figures

List of Tables

List of Symbols

The next list describes the symbols that will be frequently used within the body of the document. By default, matrices are denoted with bold capital letters, vectors are denoted with bold small letters.

Abbreviations

GP Gaussian Process

LA Laplace Approximation

LWD Logging While Drilling

MCMC Markov chain Monte Carlo

MF-ASC Multi-Fidelity Active Search with Cokriging

MWD Measurements While Drilling

ROC AUC Area Under the Curve of the Receiver Operating Characteristic

UCB Upper Confidence Bound

Math symbols in chapter 2

$\boldsymbol{\theta}_d$ Parameters of kernel k_d (a multi-dimensional real vector).

$\boldsymbol{\theta}_l$ Parameters of kernel k_l (a multi-dimensional real vector).

$\delta(\cdot)$ Latent residual Gaussian Process.

λ Log-likelihood; $\log p(\mathbf{y}^L, \mathbf{y}^H|\boldsymbol{\xi})$.

$\mathbf{X}_H$ Matrix of object-features from high-fidelity set; $\mathbf{X}_H = \{\mathbf{x}_i^H\}_{i=1}^{n_H}$.

$\mathbf{X}_L$ Matrix of object-features from low-fidelity set; $\mathbf{X}_L = \{\mathbf{x}_i^L\}_{i=1}^{n_L}$.

$\mathcal{L}$ Approximate log marginal likelihood; $\log \widetilde{q}(\mathbf{y}^L, \mathbf{y}^H|\mathbf{X}_L, \mathbf{X}_H, \rho, \boldsymbol{\theta}_l, \boldsymbol{\theta}_d)$.

Ω The measurable domain of data; $\Omega \subset \mathbb{R}^d$.

$\omega(\cdot)$ The first derivative of $\sigma(\cdot)$.

Ψ Un-normalized log-posterior over the latent variables; $\log p(\mathbf{y}^L, \mathbf{y}^H|\boldsymbol{\xi}) + \log p(\boldsymbol{\xi}|\mathbf{X}_L, \mathbf{X}_H)$.

ρ Linear coefficient for co-kriging dependency; $\rho \in \mathbb{R}$.

$\sigma(\cdot)$ Sigmoid function.

$\mathrm{diag}(\cdot)$ A function that maps a vector into the matrix with this vector on its diagonal.

$\zeta(\cdot)$ The second derivative of $\sigma(\cdot)$.

$c(\cdot)$ A binary function defined on Ω.

D_H Sample of high-fidelity data of size n_H.

D_L Sample of low-fidelity data of size n_L.

$f_H(\cdot)$ Latent Gaussian Process for high-fidelity; $f_H(\mathbf{x}) = \rho f_L(\mathbf{x}) + \delta(\mathbf{x})$.

$f_L(\cdot)$ Latent Gaussian Process for low-fidelity.

$k_d(\cdot, \cdot)$ Prior kernel for $\delta(\cdot)$.

$k_l(\cdot, \cdot)$ Prior kernel for $f_L(\cdot)$.

Math symbols in chapter 3

$\widetilde{w}(\cdot)$ internal approximation of the user relevance score (low-fidelity); $\widetilde{w} : \mathcal{X} \mapsto [0, 1]$

β_t A parameter of the exploration-exploitation tradeoff in the MF-ASC algorithm; $\beta_t \geq 0$.

Λ A total budget on queries during the active-search session; $\Lambda \in \mathbb{R}^+$.

r A ratio of low-fidelity to high-fidelity calls in the MF-ASC algorithm; $r \geq 0$.

c Cost of querying high-fidelity w; $c \in \mathbb{R}^+$.

S^H A set of evaluated items with high-fidelity function during the active-search session; $S^H \subseteq \mathcal{X}$.

$\widetilde{c}$ Cost of querying low-fidelity $\widetilde{w}$; $\widetilde{c} \in \mathbb{R}^+$.

S^L A set of evaluated items with low-fidelity function during the active-search session; $S^L \subseteq \mathcal{X}$.

$\mu(\cdot)$ Mean (expected) relevance.

$\mathcal{X}$ Dataset – a sample of items from an arbitrary set.

λ Regularization parameter for the kernel in MF-ASC.

$w(\cdot)$ User relevance score (high-fidelity); $w : \mathcal{X} \mapsto [0, 1]$

ρ Linear coefficient for co-kriging dependency between w and $\widetilde{w}$; $\rho \in \mathbb{R}$.

$\sigma(\cdot)$ Standard deviation of the relevance.

$\mathbf{K}_0$ Prior covariance matrix of items in the dataset $\mathcal{X}$.

$k_0(\cdot, \cdot)$ Prior kernel function on pairs of items in the dataset $\mathcal{X}$ that corresponds to the covariance matrix $\mathbf{K}_0$; $\mathbf{K}_0 = [k_0(x_i, x_j)]_{i,j=1}^{|\mathcal{X}|}$.

$\mathcal{U}(\cdot)$ A utility function of an item-set that measures its total relevance; $\mathcal{U}(S) = \sum_{x \in S} w(x)$.

Chapter 1

Introduction

The task of finding objects in the database that correspond to the user's interest is essential in many areas, for example: searching for web pages, patents, scientific articles, legal precedents, medical records and employee resumes. In such tasks, the search needs of the end user are usually difficult to be expressed in the form of declarative queries, moreover, the user himself may not have an explicit idea of the target result, so it can only be formed during the search process. Such conditions make interesting to research search algorithms, that actively interact with the user, choosing new objects and asking the user to evaluate them, and adapt to the interests of the user within this process taking into account his or her reaction to intermediate results.

In addition to information about the user's reaction, the system can have additional information sources about the relevance of objects, such as the history of search queries of all users, estimates of intermediate results by a virtual assistant or another user. These sources are potentially useful for improving search results and reducing load on the user. A similar approach using several sources of information has proven itself in the field of engineering optimization based on data of variable *fidelity* (precision), also called a *multi-fidelity modeling*, where a combination of *high-fidelity*, e.g. precise slow simulations, and *low-fidelity*, e.g. fast approximate models, are used to speed up the process of selecting configurations of the optimized object.

At the same time, one of the best practices in modern data analysis is a combination of human and artificial intelligence: the machine works better and faster with well-structured patterns, whereas the human helps to process rare anomalous objects or events, unstructured information, and also sets the goals for the machine data processing. This collaboration is especially important in cases of high cost of the machine error on the one hand, and high cost of human labor on the other – all this takes place in the examples of applied areas mentioned above.

Despite multi-fidelity modeling has been widely adopted in engineering optimization, the methods from this discipline are not always applicable or correct in the scenarios described above. For example, Gaussian processes are often used to model continuous response surfaces as they offer the exact analytic solution, however, if the objective function represents classes of objects instead of a real-valued quantitative characteristic, then for deriving forecasts, one should take into account the probabilities of the classes, but not their nominal values. In this case, posterior class predictions by a Gaussian process have no analytical expression even in case of a single data source. Another example relates to the search problem, which differs from the optimization problem due to the discreteness of the search space and the absence of a reward for the search algorithm for a re-selected object, which can make standard engineering optimization algorithms ineffective for the search problem.

To sum up, the topic of multi-fidelity active search for objects from databases based on modeling heterogeneous sources of information on their relevance is challenging and represents a particular research and applied interest nowadays.

The **topic** of the thesis is multi-fidelity models for classification and active search. The **subject** of the research is the methods of modeling heterogeneous data sources and methods of performing active search using such models. The **aims** of the work are to develop a multi-fidelity active search framework, research its core components, and validate them in the applied problems. These aims lead to the following **problems**:

1. to develop a computationally efficient method for modeling object classes based on heterogeneous data sources;
2. to develop an algorithm for active search of objects based on heterogeneous sources of data of their relevance;
3. to implement problem-oriented software for conducting computational experiments with the developed methods and algorithms based on data.

The main scientific novelty of this work comprises the following:

- a new Bayesian inference scheme has been developed for the classification problem based on Gaussian processes for the case when data comes from several sources with different levels of noise in the labels;
- a new active search method has been developed based on the co-kriging model of the user relevance estimates and those calculated by artificial intelligence.

The **practical utility** of the developed methods has been demonstrated in a series of industrial petroleum engineering projects, where multi-fidelity active search algorithm helped to reduce the amount of manual labor for datasets annotation for directional drilling accidents prediction, whereas multi-fidelity classification method was employed for improving quality of datasets for data-driven rock-type identification. In addition, our preliminary research of Bayesian optimization methods was successfully applied in the active muon shield optimization for the SHiP experiment at CERN.

The main defense statements are:

1. The proposed multi-fidelity classification model based on co-kriging of latent Gaussian processes is more robust to noise in low-fidelity data source and performs with high quality on practice compared to the existing alternatives.
2. The developed approximate Bayesian inference based on Laplace approximation is a computationally effective alternative to MCMC method, that provides comparable quality of predictions.
3. The proposed algorithm for multi-fidelity active search surpasses single-fidelity methods when the correlation between data sources is high by delivering more relevant results within the same budget on evaluations.
4. With the developed multi-fidelity classification and active search framework, a number of industrial problems have been solved.
5. The developed methods were included into software packages for data-driven oilfield development and neural architecture search.

The work is based on **methodology** of machine learning, surrogate modeling, and Bayesian optimization. The **reliability** of the results presented in the work is determined by the use of correct machine learning methods based on well-studied approaches from the theory of mathematical statistics, as well as the results of numerical and applied experiments.

The main part of the thesis consists of five chapters. The first chapter contains introduction of the thesis and basic concepts. A method for modeling the classes of objects in case of heterogeneous data sources and an algorithm for active search for objects by heterogeneous data sources are described in the second and the third chapters respectively. The fourth chapter demonstrates the application of the multi-fidelity active search methodology and the developed methods in several applied problems. The fifth chapter contains conclusions.

1.1 Gaussian Processes

The major part of the thesis utilizes Gaussian processes as the main method to model data. Therefore it is important to introduce them first. This section defines some basic concepts related to Gaussian processes in the context of Machine Learning based on the book [1].

A *Gaussian Process* (GP) is defined as an indexed collection of random variables over a set $\mathcal{X}$, any finite number of which has a joint Gaussian distribution. To specify this stochastic process $f(x)$, one should define its *mean function* $\mu(x) : \mathcal{X} \to \mathbb{R}$ and *covariance function* $k(x, x') : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$:

$$\mu(x) = \mathbb{E}[f(x)], \tag{1.1}$$

$$k(x, x') = \mathbb{E}[(f(x) - \mu(x))(f(x') - \mu(x'))]. \tag{1.2}$$

Values of $f(x)$ correspond to a random variable of the process at index $x \in \mathcal{X}$.

Covariance functions are also called as *kernels*, they play an important role in the distribution over functions f. The most widely used example of a kernel is the squared exponential covariance function $k_{<\eta,\theta>}(x, x') = \eta e^{-\theta||x-x'||^2}$, which has two parameters η and θ.

When some values of random variables in the Gaussian process are fixed, one can obtain the *posterior* distribution of the others, which will also be Gaussian [7] (properties of marginal and conditional distributions of multivariate Gaussian distribution):

Given

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_a \\ \mathbf{y}_b \end{bmatrix} \sim \mathcal{N}\left(\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b \end{bmatrix}, \mathbf{K} = \begin{bmatrix} \mathbf{K}_{aa} & \mathbf{K}_{ab} \\ \mathbf{K}_{ba} & \mathbf{K}_{bb} \end{bmatrix}\right) \tag{1.3}$$

The following holds:

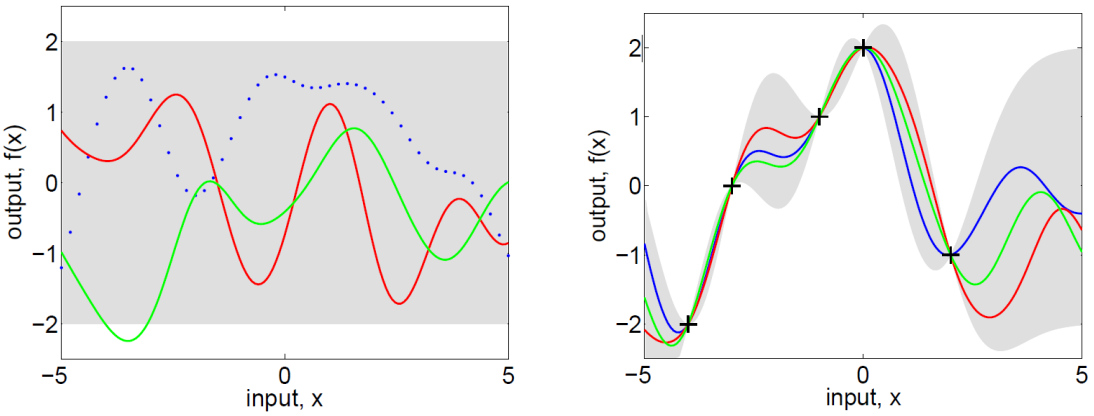
$$\mathbf{y}_i \sim \mathcal{N}(\boldsymbol{\mu}_i, \mathbf{K}_{ii}), \tag{1.4}$$

and

$$\mathbf{y}_i|\mathbf{y}_j \sim \mathcal{N}\left(\boldsymbol{\mu}_{i|j} = \boldsymbol{\mu}_i + \mathbf{K}_{ij}\mathbf{K}_{jj}^{-1}(\mathbf{y}_j - \boldsymbol{\mu}_j), \mathbf{K}_{i|j} = \mathbf{K}_{ii} - \mathbf{K}_{ij}\mathbf{K}_{jj}^{-1}\mathbf{K}_{ji}\right) \tag{1.5}$$

for $i, j \in \{a, b\}$.

Figure 1.1 gives an example of modeling a 1-dimensional function with a Gaussian process. To get more intuition on effects of various kernels and fixed values, one can play with the following interactive tool [8].



(A) Prior (B) Posterior

FIGURE 1.1: Examples of the samples from the prior and posterior Gaussian process defined on the real interval $\mathcal{X} = [-5, 5]$ from [1]. The left figure shows samples drawn from the *prior* Gaussian process with no fixed values, whereas the right figure shows samples drawn from the *posterior* Gaussian process with five fixed values in points shown as black crosses. Gray area shows 95% confidence interval of random variables' values.

In conclusion, Gaussian processes provide a probabilistic approach to kernel machine learning with several practical properties. For example: the domain knowledge of the problem can be incorporated into prior mean and kernel functions (e.g. a seasonal nature of the function can be reflected in the periodic kernel); predictions are given in the form of distribution as opposed to the point estimates; regression problems have exact analytic solutions, although for classification problems various approximations are available.

1.2 Multi-fidelity methods

The thesis is dedicated to two specific multi-fidelity methods, thus, in the current section we will introduce this field in the broader scope [2, 9].

A *high-fidelity* term henceforth indicates the property of models or data to have an acceptable accuracy (or quality of predictions in general) for the target application. In turn, a *low-fidelity* term stands for less accurate models or data, but which are cheaper to obtain with respect to some resource (typically time, energy, or money).

Multi-fidelity methods enable combinations of both low- and high- fidelity data/models to achieve trade-offs between accuracy and required resources. These methods sometimes can give superior accuracy with lower amount of resources than using low- and high-fidelity data/models separately, that is, *single-fidelity methods* [10–12].

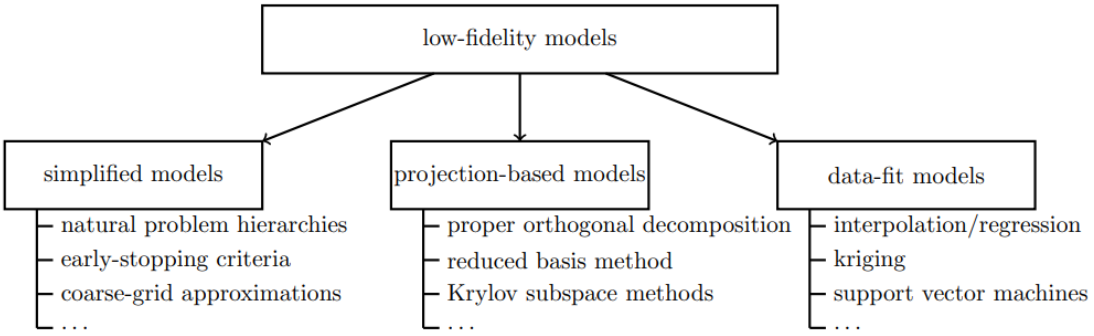


FIGURE 1.2: Types of low-fidelity models [2].

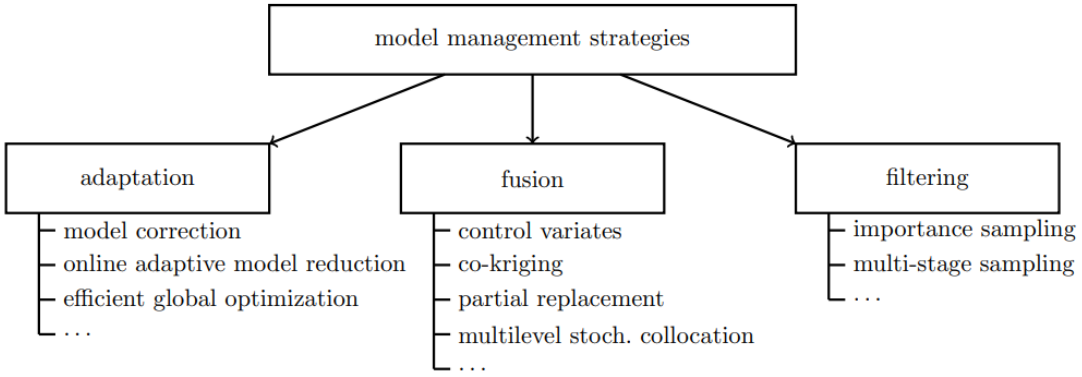


FIGURE 1.3: Types of model management strategies [2].

There are two key aspects of multi-fidelity methods: low-fidelity sources of information (models/data) and model management strategies [2].

Low-fidelity sources of information can be of various types (see Figure 1.2): simplified models are constructed out of the domain knowledge of the problem or solution implementation, projection-based models are obtained via mathematical reduction of the problem structure, and data-fit models are basically the ones trained on high-fidelity data to produce predictions for out-of-sample inputs, so these predictions are considered to be low-fidelity.

Model management strategies are responsible for balancing resources between low- and high- fidelity models as well as controlling the accuracy of the combined predictions. There are also several types of strategies (see Figure 1.3): an adaptation strategy uses high-fidelity data to improve the low-fidelity model, a fusion strategy combines low- and high-fidelity models into the one multi-fidelity model, a filtering strategy uses low-fidelity models to decide where usage of the high-fidelity model is needed.

In conclusion, there is a broad range of various types of multi-fidelity methods, they are ubiquitous in data science and engineering. These methods have been advancing over a decade, and have been adopted in numerous applications.

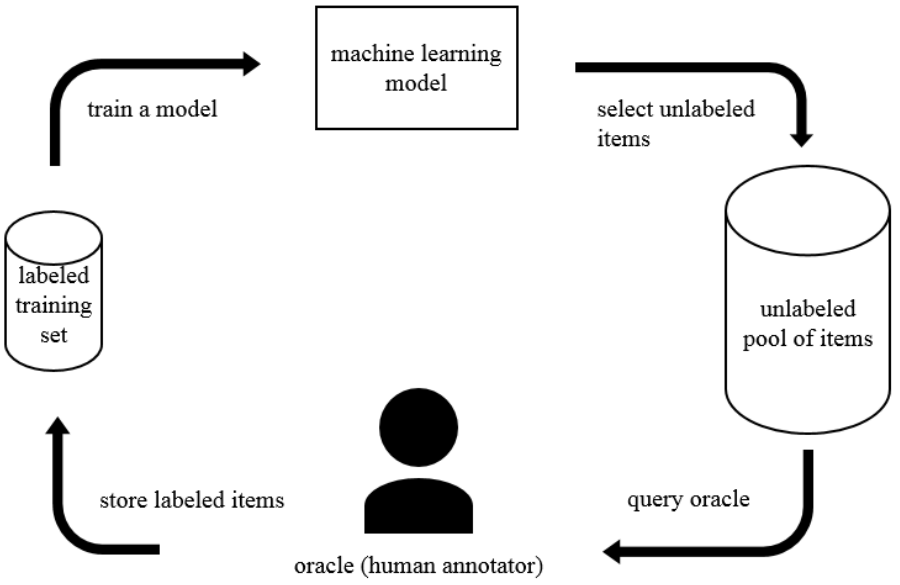


FIGURE 1.4: Active Learning cycle [3].

1.3 Bayesian optimization, Active Learning and Active Search

A substantial part of my thesis touches an Active Search [13] problem, which is a special case of Active Learning [3] and Bayesian optimization [14]. This section briefly introduces both concepts.

Active Learning [3] is an approach to training the machine-learning algorithms, during which the training set is built interactively: at first, all items have unknown labels, then at each iteration a trainee can select an item and query its label. Basically, the goal is to obtain a desired quality of the model with as few iterations as possible. The Active Learning cycle is illustrated in Figure 1.4.

Bayesian optimization is a class of methods that employ machine-learning models to solve the following black-box optimization problem:

$$\max_{x \in \mathcal{X}} f(x), \tag{1.6}$$

where $\mathcal{X}$ is a compact set, and f is a *black-box* function, which has an unknown analytic expression and a mathematical structure e.g. derivatives are not observable, no special properties like concavity etc., yet typically some assumptions are made to ensure convergence of the methods (for example, one can assume that f is a realization of the Gaussian process) [15]. During the optimization process, one can only evaluate this function at the selected elements of $\mathcal{X}$, each evaluation takes some resources. Basically, the goal is to approach the global optimum of f with as less resources as possible.

To perform Bayesian optimization of f, we need to define two components: a *surrogate model* $\hat{f}$ that approximates f based on previous evaluations as well as allows to estimate the uncertainty of approximation (typically, a Gaussian process is used), and an *acquisition criterion* a, that uses the surrogate model to decide which element of $\mathcal{X}$ is to use for next evaluating f. The general procedure of Bayesian optimization is sketched in Algorithm 1.

Algorithm 1 Bayesian optimization procedure [14]

1: **for** $t = 1, 2, ...$ **do**
2: Find x_t with the acquisition criterion $x_t = \underset{x}{\text{argmax}}\ a(x|\hat{f}_{D_{t-1}})$
3: Evaluate function f at element x_t: $y_t = f(x_t) + \epsilon_t$, where ϵ_t is noise.
4: Update the dataset $D_t = \{(x_t, y_t)\} \cup D_{t-1}$
5: Re-train the surrogate model $\hat{f}$ on the dataset D_t: $\hat{f}_{D_t}$
6: **end for**

Active Learning and Bayesian optimization have similar protocols, but their goals are different: the former aims to maximize quality of the model, whereas the latter aims to find the global optimum of the function (quality of the surrogate model is secondary). In the same fashion, a new class of problem has been recently proposed: *Active Search* [13] aims to find as many items of a given class as possible. This can also be extended to maximizing the total utility of the found items [16].

In conclusion, a lot of applications face the problem of having large unannotated datasets, whereas the most useful models require supervised training. To obtain cost-effective solutions, one can employ Bayesian optimization, Active Learning or Active Search depending on the application goals. These methods especially in the combination with multi-fidelity methods have gained a lot of research attention recently.

Chapter 2

Multi-fidelity classification

The problem of multi-fidelity modeling [2] arises in the broad range of applied disciplines, such as engineering design, medical diagnostics, and even product development, when an object of interest can be modeled with a cheaper, yet typically less reliable alternative. The main motivation behind multi-fidelity modeling is that low-fidelity data can bring additional benefits in terms of accuracy/cost trade-off, when it is used properly along with high-fidelity data [17, 18]. For example, an article [19] demonstrates that high-quality linguistic annotation results can be achieved with much lower expenses when non-expert annotators (i.e. low-fidelity data) are employed. The authors concluded that four non-experts per item were enough on average to achieve an expert-level annotation quality for their tasks, although this condition can be relaxed further, by requiring multiple annotations only for a fraction of the dataset. Similarly, in engineering design [20] a high-fidelity source of data can be a physical experiment, whereas a low-fidelity can be a mathematical model or a computer simulation.

To solve the problem of multi-fidelity classification, one can use a wide range of machine learning methods, starting from stacking-based ensemble of the classifiers [21], through the composite neural networks [22], to Bayesian neural networks [23] and Gaussian processes [24]. The final choice of the method, however, depends on the downstream application and the nature of the data. For example, in case of large-scale categorical datasets, one should choose stacking-based ensemble of the decision trees. In contrast, for small datasets with real-valued features and a need of estimating uncertainty of predictions, one needs Bayesian neural networks or Gaussian processes. The former is more flexible in modeling various dependencies of data (e.g. approximate non-stationary kernels or non-linear correlations between low- and high-fidelity values), however, such models can give over-confident predictions in out-of-sample areas, leading to worse performance of active learning (or active search) methods [25].

In this work, we propose a co-kriging model for latent low- and high- fidelity functions and extend the Laplace inference algorithm for Gaussian process classification to handle this case. The novelty of our work with respect to other existing ones is adaptation of the co-kriging model to the classification problem. This model imposes specific dependency and order on sources of data, which help the model achieve better performance than more general methods in cases when nature of data is well explained by the model. We evaluate the proposed method on three groups of datasets: artificially generated under the model assumptions, real benchmark datasets with simulated noise for low-fidelity labels and real datasets with true noise. Additionally, contribution of our work includes study of effects of budget distribution among variable fidelity sources under different noise conditions and sensitivity analysis of the proposed model to its hyperparameters.

2.1 Gaussian processes for multi-fidelity modeling

Multi-fidelity modeling based on Gaussian processes (GPs) [1] is a reasonable approach for the applications discussed above, because of the Bayesian formulation [26], which allows incorporation of the prior knowledge about the task into the prediction and makes learning on small samples more robust. The latter is especially important, since high-fidelity data typically contains just a few examples. In addition, Gaussian processes are based on kernel functions, whose hyperparameters can be selected via marginal likelihood maximization instead of grid search with cross-validation.

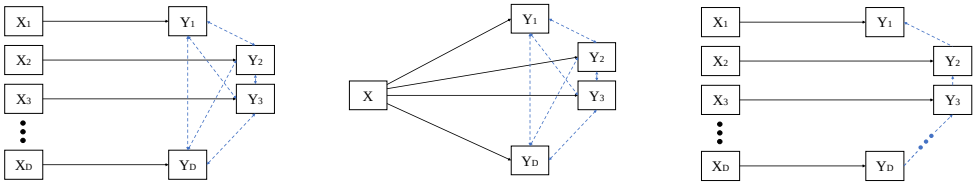
Gaussian process regression for multi-fidelity data has been thoroughly studied in recent years [27, 28], however multi-fidelity classification based on Gaussian processes has been left behind until recently. For example, the work about feasibility regions for aeroelastic stability modeling [29] pointed out that multi-fidelity methods had been limited to the continuous response models. Although discrete response models can also be approximated with continuous ones, in some extreme cases, such as binary classification, continuous approximations seem as improper as using Linear regression instead of Logistic regression. On the other hand, developing appropriate models for multi-fidelity classification is essential, because there are problems in engineering design with discrete responses. For instance, the report [30] points out the problem of a reality gap in robotic simulators and argues the importance of their ability to estimate reliability regions, where accomplishment of actions is accurately predicted by the simulator. This problem has binary responses i.e. success or fail; simulated outcomes of robot's actions are low-fidelity data, whereas observations of real executions are high-fidelity data. Furthermore, discrete responses are common and convenient when the object of interest is a human. For example, users say they either like a new feature of the product or do

TABLE 2.1: Computational complexity of approximate inference methods [5]. The running time is normalized to that of the LA method. n indicates the size of the training set, m indicates the number of inducing points [6].

	LA	EP	scalable EP [34]	scalable VI [33]	MCMC
idea	quadratic expansion around the mode	marginal moment matching	stochastic mini-batch optimization	single variational bound	sampling, thermo-dynamic integration
complexity	$O(n^3)$	$O(n^3)$	$O(m^3)$	$O(m^3)$	$O(n^3)$
running time	1	10	depends on m	depends on m	> 500
pros	speed	accuracy	speed and accuracy trade-off	speed and accuracy trade-off	theoretical accuracy
cons	underestimates predictive probabilities	slow, no convergence guarantees	no convergence guarantees	non-closed form of computations	very slow

not during A/B testing, which gives direct evidence of their attitude i.e. high-fidelity data, or users are just asked to imagine the feature and express their preferences during interviewing i.e. low-fidelity data.

A comprehensive introduction into GPs in the context of machine learning has been done previously [1]. We were guided by that book during the derivations of our algorithm. A more detailed study [5] of methods for approximate binary classification inference based on GPs demonstrates that Laplace Approximation (LA) is the fastest inference method with moderate accuracy, whereas Expectation Propagation (EP) is the most accurate, but runs approximately 10 times slower (see Table 2.1 for computational time of various methods). The study outlines that the former should be considered when the error rate is the main metric, although the latter delivers more accurate class probabilities. In addition, when labels contain a lot of noise, the authors outline that all approximation methods tend to produce similar results. More recent studies added scalability to EP and Variational Inference (VI) [31–34], yet they still have higher training times than LA in general, moreover, the work on GPU parallelization of exact Gaussian Processes inference [35] discusses a possibility of applying their technique for LA, which will further increase its computational performance. Another property that makes LA and EP distinct is convergence, which is guaranteed for LA in case of likelihood log-concavity and not guaranteed for EP, so the latter may face numerical optimization problems. The work [36] has also suggested using LA solution as a starting point for EP in order to improve convergence on practice and reduce the amount of EP iterations. To sum up, despite modern research has been prevalently focused on EP and VI methods, LA is still a reasonable option due to its convergence properties and computational performance.



(A) Multi-task model (B) Multi-output model (C) Multi-fidelity model

FIGURE 2.1: Differences among various cases of multi-task models. $\mathbf{X}_i$ indicates ith input set, Y_i indicates ith output set, black lines show dependency, blue dashed lines show correlations.

The supervised classification in the presence of noise in labels [37] has been studied with class-conditional random Bernoulli noise, such classification problems have also got theoretical justification of their learnability.

The prior works extensively cover the topics connected to multi-task learning [38, 39] and multi-output GPs [40]. Multi-fidelity regression based on GPs is a particular case of multi-task GPs with additional assumptions on outputs dependency structure (see Figure 2.1), which reduce computational complexity of training and inference. Multi-fidelity models were studied in a number of works [41–44], including a co-kriging setup for fidelities with an exact inference schema for their regression [27]. In our work, we adopt co-kriging for the classification problem by applying this setup on latent functions. Note that there is no exact inference schema for the GP classification for single-fidelity case, nor for the multi-fidelity one. Several recent works are dedicated to close problems, yet they all consider different aspects. For example, the work on the multivariate generalized linear geostatistical model with spatially structured bias [45] is close to ours, however, the model studied there doesn't take into account a scaling factor and the proposed inference is confined to MCMC method. A more recent work proposed a framework for handling heterogeneous outputs of GPs with stochastic variational inference [24], also there is a study of the application of heterogeneous multivariate GPs for joint species distribution modeling [46]. Compared to them our work is about a more specific model of multivariate GPs, that can be adapted to a classical algorithmic framework [1] without additional approximation techniques. This tailoring makes our method more robust to noise in labels and accurate than others that use more general models, as we show further in the experimental section.

2.2 Multi-fidelity classification with Gaussian processes

2.2.1 Problem statement

There is a binary function $c : \Omega \to \{0, 1\}$ defined on the measurable set $\Omega \subset \mathbb{R}^d$. We have two samples:

$$D_H = \{(\mathbf{x}_i^H, y_i^H)\}_{i=1}^{n_H} \text{ and } D_L = \{(\mathbf{x}_i^L, y_i^L)\}_{i=1}^{n_L}, \tag{2.1}$$

where $\mathbf{x}_i^L, \mathbf{x}_i^H \in \Omega$ and $y_i^L, y_i^H \in \{0, 1\}$. Let us also denote $\mathbf{X}_L = \{\mathbf{x}_i^L\}_{i=1}^{n_L}$, and $\mathbf{X}_H = \{\mathbf{x}_i^H\}_{i=1}^{n_H}$.

Sample D_H contains high-fidelity data, that is, much more reliable labels than D_L, which contains low-fidelity data respectively, so its labels can be biased and noisier. Using the Bayesian approach we formally express this assumption with the following model:

$$\begin{aligned} &c(x) = \mathbb{I}\left[f_H(x) > 0\right], \\ &p\left(y_i^H = 1 | f_H(\mathbf{x}_i^H)\right) = \sigma\left(f_H(\mathbf{x}_i^H)\right), \\ &p\left(y_i^L = 1 | f_L(\mathbf{x}_i^L)\right) = \sigma\left(f_L(\mathbf{x}_i^L)\right), \end{aligned} \tag{2.2}$$

where $\mathbb{I}$ is an indicator function; $\sigma(z) = \frac{1}{1+exp(-z)}$ is a sigmoid function; f_L and f_H are Gaussian processes on Ω. In our model we assume these processes to be dependent via co-kriging model [27]:

$$f_H(\mathbf{x}_i^H) = \rho f_L(\mathbf{x}_i^H) + \delta(\mathbf{x}_i^H), \tag{2.3}$$

where $\rho \in \mathbb{R}$ is a linear coefficient, and δ is a residual Gaussian process independent of f_L. Processes f_L and δ have prior kernels k_l and k_d with hyper-parameters $\boldsymbol{\theta}_l$ and $\boldsymbol{\theta}_d$ respectively. Such dependency between latent processes has been on the one hand acknowledged in many engineering applications [47], on the other hand, it corresponds to the optimal estimate of high-fidelity data according to the Theorem on normal correlation (see [48], theorem 13.1). Parameter ρ can reduce or increase the confidence of the high-fidelity model compared to the low-fidelity one, in particular, $\rho = 1$ corresponds to the case when the low-fidelity source contains high-fidelity labels with additive noise. This parameter is also useful for the cases, when low- and high- fidelity labels are mostly opposed to each other. Gaussian process δ can compensate predictions for input-dependent bias in low-fidelity data.

Finally, assuming (2.2) and (2.3), which we will call a *latent co-kriging model*, we would like to train a classifier $\hat{c}$ that estimates the function c using samples (2.1).

2.2.2 Solution

For simplicity of notation we omit specifying hyper-parameters (ρ and parameters of kernels $\boldsymbol{\theta}_l$, $\boldsymbol{\theta}_d$) as conditions of probabilities in formulas below.

The predictive distribution of f_H at $\mathbf{x}_* \in \Omega$ is:

$$p(f_*^H|D_L, D_H, \mathbf{x}_*) = \iint p(f_*^H|\mathbf{f}^L, \boldsymbol{\delta}, \mathbf{X}_L, \mathbf{X}_H, \mathbf{x}_*)p(\mathbf{f}^L, \boldsymbol{\delta}|D_L, D_H)d\mathbf{f}^L d\boldsymbol{\delta}, \tag{2.4}$$

where

$$\begin{aligned}\boldsymbol{\delta} &= \left(\delta(\mathbf{x}_1^H), ..., \delta(\mathbf{x}_{n_h}^H)\right)^\mathsf{T}, \\ \mathbf{f}^L &= \left(f^L(\mathbf{x}_1^L), ..., f^L(\mathbf{x}_{n_L}^L), f^L(\mathbf{x}_1^H), ..., f^L(\mathbf{x}_{n_h}^H)\right)^\mathsf{T}.\end{aligned}$$

The probability of c to be 1 at point $\mathbf{x}_*$ can be expressed by marginalization of the predictive distribution:

$$p(c(\mathbf{x}_*) = 1|D_L, D_H, \mathbf{x}_*) = \int \sigma(f_*^H)p(f_*^H|D_L, D_H, \mathbf{x}_*)df_*^H. \tag{2.5}$$

Integrals (2.4) and (2.5) don't have analytic solutions, therefore they have to be numerically integrated or approximated analytically. In this work we use Laplace Approximation method to handle the former, whereas the predicted class label based on the latter integral can be easily calculated in the binary case once the predictive distribution p is estimated by a Gaussian q ([49], Section 10.3):

$$\begin{aligned}\hat{c}(\mathbf{x}_*) &= \mathbb{I}\left[\int \sigma(f_*^H)q(f_*^H|D_L, D_H, \mathbf{x}_*)df_*^H > \frac{1}{2}\right] = \\ &= \mathbb{I}\left[\int f_*^H q(f_*^H|D_L, D_H, \mathbf{x}_*)df_*^H > 0\right].\end{aligned} \tag{2.6}$$

The last equality in (2.6) takes place due to the asymmetric property of $\sigma(x) - \frac{1}{2}$ and Gaussian (symmetric) property of q. Note: this equality holds only for indicators, but does not hold for the integrals in general.

2.2.2.1 Laplace Approximation

Prediction based on GPs requires two steps [1]:

1. Obtaining a latent predictive distribution for the test point via marginalizing the posterior distribution over all possible latent values at training points;

2. Marginalizing it over all possible latent values at the test point in order to produce a probabilistic prediction.

Unlike the regression problem, where marginalizations are straightforward because all underlying components are Gaussian, prediction of classes is analytically intractable due to non-Gaussian likelihoods.

The idea of Laplace's method is to handle intractability at step 1 by applying a second order Taylor expansion of posterior's logarithm around its maximum. Thus, we obtain a Gaussian approximation of the posterior distribution, which, in turn, makes approximate predictive distribution to be also Gaussian. Next, intractability of step 2 can be resolved by replacing marginalization with maximum a posteriori predictions [49] or can be approximated with numerical techniques [50, 51].

In the next three sections 2.2.2.2, 2.2.2.3 and 2.2.2.4 we will adjust our solution to fit the algorithmic framework for Laplace Approximation. The key challenge in our case is dependence of y_i^H on multiple latent components, which requires substantial modifications of basic algorithms.

2.2.2.2 Mode-fitting

The posterior distribution in integral (2.4) is approximated with Gaussian distribution $q(\cdot)$:

$$p(\mathbf{f}^L, \boldsymbol{\delta}|D_L, D_H) \approx q(\mathbf{f}^L, \boldsymbol{\delta}|D_L, D_H) = \mathcal{N}\left(\boldsymbol{\xi} = \begin{bmatrix}\mathbf{f}^L \\ \boldsymbol{\delta}\end{bmatrix} \;\middle|\; \hat{\boldsymbol{\xi}}, \boldsymbol{\Sigma}^{-1}\right), \tag{2.7}$$

where $\boldsymbol{\Sigma} = -\nabla\nabla \log p(\boldsymbol{\xi}|D_L, D_H)\big|_{\boldsymbol{\xi}=\hat{\boldsymbol{\xi}}}$ and $\hat{\boldsymbol{\xi}} = \underset{\boldsymbol{\xi}}{\operatorname{argmax}}\; p(\boldsymbol{\xi}|D_L, D_H)$. Thus, for obtaining approximate posterior distribution we need to calculate these parameters.

According to Bayes formula and monotonic increase of log function, the problem of finding $\hat{\boldsymbol{\xi}}$ is equivalent to:

$$\underset{\boldsymbol{\xi}}{\operatorname{argmax}}\; p(\boldsymbol{\xi}|D_L, D_H) = \underset{\boldsymbol{\xi}}{\operatorname{argmax}}\; \left[\log p(\mathbf{y}^L, \mathbf{y}^H|\boldsymbol{\xi}) + \log p(\boldsymbol{\xi}|\mathbf{X}_L, \mathbf{X}_H)\right], \tag{2.8}$$

where $\mathbf{y}^L = \left(y_1^L, ..., y_{n_L}^L\right)^\mathsf{T}$ and $\mathbf{y}^H = \left(y_1^H, ..., y_{n_H}^H\right)^\mathsf{T}$. Note that the probability of evidence is omitted, since it is independent of the argument. The problem (2.8) has a unique solution, see details in the section 2.2.3.

Let us now define $\Psi(\boldsymbol{\xi}) \stackrel{\triangle}{=} \log p(\mathbf{y}^L, \mathbf{y}^H|\boldsymbol{\xi}) + \log p(\boldsymbol{\xi}|\mathbf{X}_L, \mathbf{X}_H)$, and look at its components in more detail. Let us also introduce $\mathbf{X} = \mathbf{X}_L \cup \mathbf{X}_H$.

The prior distribution of $\boldsymbol{\xi}$ is normal:

$$p(\boldsymbol{\xi}|\mathbf{X}_L, \mathbf{X}_H) \sim \mathcal{N}\left(0, K = \begin{bmatrix} k_l(\mathbf{X}, \mathbf{X}) & 0 \\ 0 & k_d(\mathbf{X}_H, \mathbf{X}_H) \end{bmatrix}\right). \tag{2.9}$$

Log-likelihood is:

$$\lambda \triangleq \log p(\mathbf{y}^L, \mathbf{y}^H|\boldsymbol{\xi}) = \sum_{i=1}^{n_l} \log\ \sigma\big(\widetilde{y}_i^L f^L(\mathbf{x}_i^L)\big) + \sum_{i=1}^{n_h} \log\ \sigma\big(\widetilde{y}_i^H(\rho f^L(\mathbf{x}_i^H) + \delta_i)\big),$$

where for simplicity of notation we use:

$$\delta_i = \delta(\mathbf{x}_i^H),\ \widetilde{y}_i^L = (2y_i^L - 1), \text{ and } \widetilde{y}_i^H = (2y_i^H - 1).$$

Having figured out expressions for components of $\Psi(\boldsymbol{\xi})$, the solution of problem (2.8) can be found with iterative Newton's method:

$$\hat{\boldsymbol{\xi}}^{\text{ new}} = \hat{\boldsymbol{\xi}}^{\text{ old}} - (\nabla\nabla\Psi)^{-1}\nabla\Psi\big|_{\boldsymbol{\xi}=\hat{\boldsymbol{\xi}}^{\text{ old}}}.$$

2.2.2.3 Model selection

Let us denote $\widetilde{q}(.)$ a Gaussian approximation of the marginal likelihood

$$p(\mathbf{y}^L, \mathbf{y}^H|\mathbf{X}_L, \mathbf{X}_H, \rho, \boldsymbol{\theta}_l, \boldsymbol{\theta}_d).$$

The model selection implies finding hyper-parameters ρ, $\boldsymbol{\theta}_l$, and $\boldsymbol{\theta}_d$ that maximize the approximate log marginal likelihood (this approximation is obtained similarly to the single-fidelity case [1]):

$$\mathcal{L} \triangleq \log \widetilde{q}(\mathbf{y}^L, \mathbf{y}^H|\mathbf{X}_L, \mathbf{X}_H, \rho, \boldsymbol{\theta}_l, \boldsymbol{\theta}_d) = -\frac{1}{2}\hat{\boldsymbol{\xi}}^{\mathsf{T}}\mathbf{K}^{-1}\hat{\boldsymbol{\xi}} + \lambda - \frac{1}{2}\log|\mathbf{B}|, \tag{2.10}$$

where $\mathbf{B} = \mathbf{I} + \mathbf{W}^{\frac{1}{2}}\mathbf{K}\mathbf{W}^{\frac{1}{2}}$ and

$$\mathbf{W} \triangleq -\nabla\nabla_{\boldsymbol{\xi}} \log p(\mathbf{y}^L, \mathbf{y}^H|\boldsymbol{\xi}) = \begin{bmatrix} \mathbf{A} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \rho^2\mathbf{D} & \rho\mathbf{D} \\ \mathbf{0} & \rho\mathbf{D} & \mathbf{D} \end{bmatrix}; \tag{2.11}$$

$$\mathbf{A} = \nabla\nabla_{f^L(\mathbf{X}_L)}\lambda = \operatorname{diag}\left(\left[\omega\left(f^L(\mathbf{x}_i^L)\right)\right]_{i=1}^{n_l}\right),$$
$$\mathbf{D} = \nabla\nabla_{\boldsymbol{\delta}}\lambda = \operatorname{diag}\left(\left[\omega\left(\rho f^L(\mathbf{x}_i^H) + \delta_i\right)\right]_{i=1}^{n_h}\right),$$
$$\omega(z) = \sigma'(z) = \sigma(z)(1-\sigma(z)),$$

where $\operatorname{diag}(\cdot)$ is a function that maps a vector into the matrix with this vector on its diagonal.

Unlike single-fidelity case, $\mathbf{W}$ in multi-fidelity case is non-diagonal, so computation of its square root is not straightforward. We have derived the exact formula for its fast and numerically stable calculation:

$$\mathbf{W}^{\frac{1}{2}} = \begin{bmatrix} \mathbf{A}^{\frac{1}{2}} & 0 \\ 0 & \frac{1}{\sqrt{\rho^2+1}}\begin{bmatrix}\rho^2 & \rho \\ \rho & 1\end{bmatrix} \otimes \mathbf{D}^{\frac{1}{2}} \end{bmatrix}, \tag{2.12}$$

note that matrices $\mathbf{A}$ and $\mathbf{D}$ are diagonal, so their square roots are easily calculated.

In order to maximize log marginal likelihood (2.10), one can use gradient-based optimization, which requires its partial derivatives w.r.t. hyper-parameters. The rest of this subsection is dedicated to analytic derivation of computationally effective formilas of partial derivatives.

Partial derivatives w.r.t. $\boldsymbol{\theta}_l$ and $\boldsymbol{\theta}_d$.

Derivatives of $\mathcal{L}$ and $\hat{\boldsymbol{\xi}}$ w.r.t. kernel hyper-parameters $\boldsymbol{\theta}_l$ and $\boldsymbol{\theta}_d$ are analogous to formulas in the single-fidelity case ([1], section 5.5.1), thus, we omit them here, except the formula 5.23 from [1] for partial derivatives of $\mathcal{L}$ w.r.t. components of $\hat{\boldsymbol{\xi}}$, which reduces calculation of trace to multiplication of i-th diagonal elements. That reduction doesn't take place in multi-fidelity case, since $\frac{\partial \mathbf{W}}{\partial \hat{\xi}_i}$ is not diagonal in general. We propose the following modification of that formula:

$$\frac{\partial \mathcal{L}}{\partial \hat{\xi}_i} = -\frac{1}{2}\operatorname{tr}\left((\mathbf{K}^{-1}+\mathbf{W})^{-1}\frac{\partial \mathbf{W}}{\partial \hat{\xi}_i}\right) = -\frac{1}{2}\sum_{\text{all elements}}\left((\mathbf{K}^{-1}+\mathbf{W})^{-1} \circ \frac{\partial \mathbf{W}}{\partial \hat{\xi}_i}\right), \tag{2.13}$$

where $\circ$ is an Hadamard (entrywise) product. Note that $\frac{\partial \mathbf{W}}{\partial \hat{\xi}_i}$ is a sparse matrix that has at most 4 non-zero elements, therefore computation time of the derivatives remains linear (see details in equations (2.14), (2.15), (2.16)).

To describe components of (2.13), let us denote $\mathbf{M} \triangleq (\mathbf{K}^{-1} + \mathbf{W})^{-1}$.

For indices i corresponding to low-fidelity data on $\mathbf{X}_L$ $(i = 1...n_l)$:

$$\mathbf{M}_{i,i} \frac{\partial^3}{\partial \hat{\xi}_i} \lambda \equiv \mathbf{M}_{i,i} \zeta(f_L(\mathbf{x}_i^L)) \tag{2.14}$$

For indices i corresponding to low-fidelity data on $\mathbf{X}_H$ $(i = n_l + 1...n_l + n_h)$:

$$\begin{aligned} &\left(\mathbf{M}_{i,i} \frac{\partial^3}{\partial \hat{\xi}_i^3} + 2\mathbf{M}_{i,i+n_h} \frac{\partial^3}{\partial \hat{\xi}_{i+n_h} \partial \hat{\xi}_i^2} + \mathbf{M}_{i+n_h,i+n_h} \frac{\partial^3}{\partial \hat{\xi}_{i+n_h}^2 \partial \hat{\xi}_i} \right) \lambda \equiv \\ &\equiv \left(\mathbf{M}_{i,i} \rho^3 + 2\mathbf{M}_{i,i+n_h} \rho^2 + \mathbf{M}_{i+n_h,i+n_h} \rho \right) \zeta(\rho f^L(\mathbf{x}_{i-n_l}^H) + \delta_{i-n_l}) \end{aligned} \tag{2.15}$$

For indices i corresponding to delta on $\mathbf{X}_H$ $(i = n_l + n_h + 1...n_l + 2n_h)$:

$$\begin{aligned} &\left(\mathbf{M}_{i,i} \frac{\partial^3}{\partial \hat{\xi}_i^3} + 2\mathbf{M}_{i,i-n_h} \frac{\partial^3}{\partial \hat{\xi}_{i-n_h} \partial \hat{\xi}_i^2} + \mathbf{M}_{i-n_h,i-n_h} \frac{\partial^3}{\partial \hat{\xi}_{i-n_h}^2 \partial \hat{\xi}_i} \right) \lambda \equiv \\ &\equiv \left(\mathbf{M}_{i,i} + 2\mathbf{M}_{i,i-n_h} \rho + \mathbf{M}_{i-n_h,i-n_h} \rho^2 \right) \zeta(\rho f^L(\mathbf{x}_{i-n_l-n_h}^H) + \delta_{i-n_l-n_h}) \end{aligned} \tag{2.16}$$

Partial derivatives w.r.t. ρ.

Now let's look at the derivative of $\mathcal{L}$ w.r.t to ρ, which is:

$$\frac{\partial \mathcal{L}}{\partial \rho} = -\hat{\boldsymbol{\xi}}^\mathsf{T} \mathbf{K}^{-1} \frac{\partial \hat{\boldsymbol{\xi}}}{\partial \rho} + \frac{\partial \lambda}{\partial \rho} - \frac{1}{2} \frac{\partial \log |\mathbf{B}|}{\partial \rho}. \tag{2.17}$$

Note that in our setup $\mathbf{K}$ doesn't depend on ρ. Further, we will analyze the additive components of (2.17) in more detail.

To obtain the first component, we differentiate by ρ the necessary condition of the maximum $\nabla \Psi(\boldsymbol{\xi})|_{\boldsymbol{\xi}=\hat{\boldsymbol{\xi}}} = 0$, where $\nabla \Psi(\boldsymbol{\xi}) = \nabla_{\boldsymbol{\xi}} \lambda - \mathbf{K}^{-1} \boldsymbol{\xi}$, obtaining an equation on $\boldsymbol{\xi}$:

$$\begin{aligned} &\frac{\partial \hat{\boldsymbol{\xi}}}{\partial \rho} = \mathbf{K} \left(-\mathbf{W} \frac{\partial \hat{\boldsymbol{\xi}}}{\partial \rho} + \left. \frac{\partial \nabla_{\boldsymbol{\xi}} \lambda \big|_{\boldsymbol{\xi}=\hat{\boldsymbol{\xi}}}}{\partial \rho} \right|_{\text{explicit}} \right) \Rightarrow \\ &\Rightarrow \frac{\partial \hat{\boldsymbol{\xi}}}{\partial \rho} = (\mathbf{I} + \mathbf{K}\mathbf{W})^{-1} \mathbf{K} \left(\left. \frac{\partial \nabla_{\boldsymbol{\xi}} \lambda \big|_{\boldsymbol{\xi}=\hat{\boldsymbol{\xi}}}}{\partial \rho} \right|_{\text{explicit}} \right), \end{aligned} \tag{2.18}$$

where the components of the *explicit* term in formula (2.18) and derivatives of λ w.r.t. components of $\boldsymbol{\xi}$ are provided in Table 2.2.

The second component of (2.17) is:

TABLE 2.2: Components of $\boldsymbol{\xi}$, corresponding derivatives of λ and the explicit term in (2.18); here $f_i^L = f^L(\mathbf{x}_i^L)$ and $f_i^H = \rho f^L(\mathbf{x}_i^H) + \delta(\mathbf{x}_i^H)$.

components of $\boldsymbol{\xi}$	components of $\nabla_{\boldsymbol{\xi}}\lambda$	components of $\frac{\partial \nabla_{\boldsymbol{\xi}}\lambda|_{\boldsymbol{\xi}=\hat{\boldsymbol{\xi}}}}{\partial\rho}\big\|_{\text{explicit}}$
$f^L(\mathbf{X}_L)$	$y_i^L - \sigma(f_i^L)$	0
$f^L(\mathbf{X}_H)$	$\rho(y_i^H - \sigma(f_i^H))$	$y_i^H - \sigma(f_i^H) - \rho f^L(\mathbf{x}_i^H)\omega(f_i^H)$
$\delta(\mathbf{X}_H)$	$y_i^H - \sigma(f_i^H)$	$-f^L(\mathbf{x}_i^H)\omega(f_i^H)$

$$\frac{\partial\lambda}{\partial\rho} = \sum_{i=1}^{n_h} \widetilde{y}_i^H f^L(\mathbf{x}_i^H)\left(1 - \sigma\left(\widetilde{y}_i^H(\rho f^L(\mathbf{x}_i^H) + \delta(\mathbf{x}_i^H))\right)\right) + \sum_i \frac{\partial\lambda}{\partial\xi_i}\frac{\partial\xi_i}{\partial\rho}. \tag{2.19}$$

The third component is:

$$\frac{\partial \log|\mathbf{B}|}{\partial\rho} = \sum_{\text{all elements}} \left((\mathbf{K}^{-1} + \mathbf{W})^{-1} \circ \frac{\partial\mathbf{W}}{\partial\rho}\right), \tag{2.20}$$

where

$$\frac{\partial\mathbf{W}}{\partial\rho} = \begin{bmatrix} 0 & 0 \\ 0 & \begin{bmatrix}\rho^2 & \rho \\ \rho & 1\end{bmatrix} \otimes \frac{\partial\mathbf{D}}{\partial\rho}\Big|_{\text{explicit}} + \begin{bmatrix}2\rho & 1 \\ 1 & 0\end{bmatrix} \otimes \mathbf{D} \end{bmatrix} + \sum_i \frac{\partial\mathbf{W}}{\partial\xi_i}\frac{\partial\xi_i}{\partial\rho};$$
$$\frac{\partial\mathbf{D}}{\partial\rho}\Big|_{\text{explicit}} = \text{diag}\left(\left[f^L(\mathbf{x}_i^H)\zeta(f_i^H)\right]_{i=1}^{n_h}\right) \text{ and } \zeta(x) = \sigma''(x).$$

The derivation for the equation (2.20) is provided below:

$$\begin{aligned}
&\frac{\partial \log|\mathbf{B}|}{\partial\rho} = \text{tr}\left(\mathbf{B}^{-1}\frac{\partial\mathbf{B}}{\partial\rho}\right) = \text{tr}\left(\mathbf{B}^{-1}\left(\frac{\partial\mathbf{W}^{\frac{1}{2}}}{\partial\rho}\mathbf{K}\mathbf{W}^{\frac{1}{2}} + \mathbf{W}^{\frac{1}{2}}\mathbf{K}\frac{\partial\mathbf{W}^{\frac{1}{2}}}{\partial\rho}\right)\right) = \\
&= \text{tr}\left(\mathbf{B}^{-1}\left(\mathbf{W}^{-\frac{1}{2}}\mathbf{W}^{\frac{1}{2}}\right)\frac{\partial\mathbf{W}^{\frac{1}{2}}}{\partial\rho}\mathbf{K}\mathbf{W}^{\frac{1}{2}}\right) + \text{tr}\left(\mathbf{B}^{-1}\mathbf{W}^{\frac{1}{2}}\mathbf{K}\frac{\partial\mathbf{W}^{\frac{1}{2}}}{\partial\rho}\left(\mathbf{W}^{\frac{1}{2}}\mathbf{W}^{-\frac{1}{2}}\right)\right) = \\
&= \text{tr}\left(\mathbf{K}\mathbf{W}^{\frac{1}{2}}\mathbf{B}^{-1}\mathbf{W}^{-\frac{1}{2}}\left(\mathbf{W}^{\frac{1}{2}}\frac{\partial\mathbf{W}^{\frac{1}{2}}}{\partial\rho}\right)\right) + \text{tr}\left(\mathbf{W}^{-\frac{1}{2}}\mathbf{B}^{-1}\mathbf{W}^{\frac{1}{2}}\mathbf{K}\left(\frac{\partial\mathbf{W}^{\frac{1}{2}}}{\partial\rho}\mathbf{W}^{\frac{1}{2}}\right)\right) = \\
&= \text{tr}\left((\mathbf{K}^{-1} + \mathbf{W})^{-1}\frac{\partial\mathbf{W}}{\partial\rho}\right) = \sum_{\text{all elements}}\left((\mathbf{K}^{-1} + \mathbf{W})^{-1} \circ \frac{\partial\mathbf{W}}{\partial\rho}\right)
\end{aligned} \tag{2.21}$$

Whereas the last line of (2.21) is obtained because of the following identities:

$$\frac{\partial \mathbf{W}^{\frac{1}{2}}}{\partial \rho}\mathbf{W}^{\frac{1}{2}} + \mathbf{W}^{\frac{1}{2}}\frac{\partial \mathbf{W}^{\frac{1}{2}}}{\partial \rho} = \frac{\partial \mathbf{W}^{\frac{1}{2}}\mathbf{W}^{\frac{1}{2}}}{\partial \rho} = \frac{\partial \mathbf{W}}{\partial \rho} \tag{2.22}$$

$$\mathbf{K}\mathbf{W}^{\frac{1}{2}}\mathbf{B}^{-1}\mathbf{W}^{-\frac{1}{2}} = \mathbf{K}\left(\mathbf{W}^{\frac{1}{2}}\mathbf{B}^{-1}\mathbf{W}^{\frac{1}{2}}\right)\mathbf{W}^{-1} = \mathbf{K}(\mathbf{K}+\mathbf{W}^{-1})^{-1}\mathbf{W}^{-1} = (\mathbf{K}^{-1}+\mathbf{W})^{-1} \tag{2.23}$$

$$\mathbf{W}^{-\frac{1}{2}}\mathbf{W}^{\frac{1}{2}}\mathbf{B}^{-1}\mathbf{K} = \mathbf{W}^{-1}\left(\mathbf{W}^{\frac{1}{2}}\mathbf{B}^{-1}\mathbf{W}^{\frac{1}{2}}\right)\mathbf{K} = \mathbf{W}^{-1}(\mathbf{K}+\mathbf{W}^{-1})^{-1}\mathbf{K} = (\mathbf{K}^{-1}+\mathbf{W})^{-1} \tag{2.24}$$

2.2.2.4 Predictions

Once we know the estimates of parameters and hyper-parameters, we can use an ordinary schema of exact multi-fidelity posterior from [27] to obtain *MAP predictions*:

$$\mathbb{E}[f_*|D_L, D_H, \mathbf{x}_*] \approx \mathbb{E}_q[f_*|D_L, D_H, \mathbf{x}_*] = \widetilde{\mathbf{k}}_*^{\mathsf{T}}\widetilde{\mathbf{K}}^{-1}\hat{\mathbf{f}}, \tag{2.25}$$

where

$$\begin{aligned}
\widetilde{\mathbf{k}}_*^{\mathsf{T}} &= \begin{bmatrix} k_l(\mathbf{x}_*, \mathbf{X}_L) & \rho^2 k_l(\mathbf{x}_*, \mathbf{X}_H) + k_d(\mathbf{x}_*, \mathbf{X}_H) \end{bmatrix}, \\
\hat{\boldsymbol{\xi}} &= \begin{bmatrix} \hat{f}^L(\mathbf{X}_L) & \hat{f}^L(\mathbf{X}_H) & \hat{\delta}(\mathbf{X}_H) \end{bmatrix}^{\mathsf{T}}, \\
\widetilde{\mathbf{K}} &= \begin{bmatrix} k_l(\mathbf{X}_L, \mathbf{X}_L) & \rho k_l(\mathbf{X}_L, \mathbf{X}_H) \\ \rho k_l(\mathbf{X}_H, \mathbf{X}_L) & \rho^2 k_l(\mathbf{X}_H, \mathbf{X}_H) + k_d(\mathbf{X}_H, \mathbf{X}_H) \end{bmatrix}, \\
\hat{\mathbf{f}} &= \begin{bmatrix} \hat{f}^L(\mathbf{X}_L) \\ \rho\hat{f}^L(\mathbf{X}_H) + \hat{\delta}(\mathbf{X}_H) \end{bmatrix}.
\end{aligned}$$

2.2.3 Correctness of the method

Optimization problem (2.8) has a unique solution if Ψ is concave. We prove it by showing that Hessian of Ψ is negative semi-definite. The Hessian is

$$\nabla\nabla\Psi(\boldsymbol{\xi}) = -\mathbf{W} - \mathbf{K}^{-1}, \tag{2.26}$$

where K is positive semi-definite, since it is a kernel matrix.

Matrices $\mathbf{A}$ and $\mathbf{D}$ in the definition of $\mathbf{W}$ (2.11) are positive semi-definite, because their diagonal elements are non-negative. The block of $\mathbf{W}$ that contains $\mathbf{D}$ can be represented via Kronecker product:

$$\begin{bmatrix} \rho^2\mathbf{D} & \rho\mathbf{D} \\ \rho\mathbf{D} & \mathbf{D} \end{bmatrix} = \begin{bmatrix} \rho^2 & \rho \\ \rho & 1 \end{bmatrix} \otimes \mathbf{D}. \tag{2.27}$$

Both multiplicands in (2.27) are positive semi-definite, thus, their Kronecker product is also positive semi-definite [52].

Hence, matrix $\mathbf{W}$ is positive semi-definite, because it factorizes into two positive semi-definite blocks. Finally, the Hessian is negative semi-definite as a negation of the sum of two positive semi-definite matrices.

2.2.4 Experiments

We compared our model with a number of baseline approaches. The baselines are built upon ordinary Gaussian Process Classifier (**gpc**), Logistic Regression (**logit**) and Gradient Boosting Classifier (**xgb**). We trained those baselines in three modes:

1. Training only on high-fidelity data (no prefix);
2. Training on concatenated high- and low- fidelity data (with prefix **C**);
3. Stacking low-fidelity predictions, that is, predictions of a classifier trained on low-fidelity data were used as additional features for training the classifier on high-fidelity data (with prefix **S**).

All GPs-based methods used isotropic RBF kernel.

2.2.4.1 Evaluation metrics

In order to compare performance of various methods we use areas under receiver operating characteristic curves [53] (ROC AUC) metric.

Further, to aggregate performance across many tests and datasets, we average ROC AUC over them. We also supplement results with figures of *ROC AUC profiles*, which show the share of tests where the corresponding methods had greater ROC AUC than the threshold pointed on the abscissa axis. The rule of thumb for assessing such profiles is that the higher the curve, the better the corresponding method.

2.2.4.2 Datasets

We evaluated models on three groups of datasets:

1. Artificial datasets[1]: we constructed datasets by virtue of the model (2.2) and (2.3). Latent functions f_L and δ were generated as instances of Gaussian processes, linear coefficients ρ were adjusted to the desired discrepancy (noise level) between low- and high- fidelities. We used input dimensions 2, 5, 10, and 20. For each of them, we generated 10 datasets.

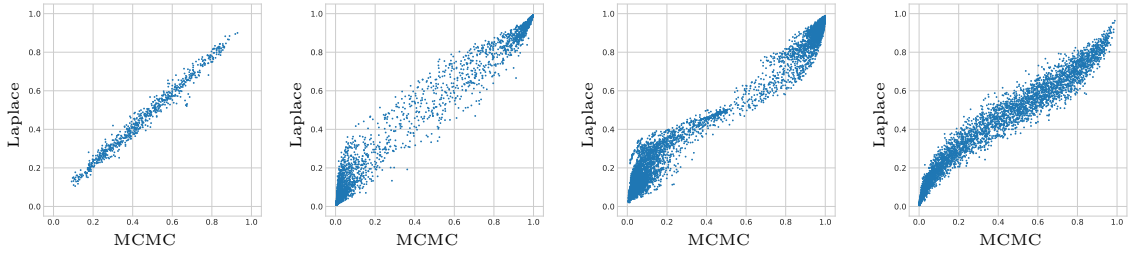
2. Datasets from Penn Machine Learning Benchmarks repository [54]: we selected several representative benchmarks with different types of features, namely `diabetes` (dbts), `german` (grmn), `waveform-40` (wvfr), `satimage` (stmg), `splice` (splc), `spambase` (spmb), `hypothyroid` (hpth), and `mushroom` (mshr). Since some datasets had multiple classes, we also selected one target representative class to test its classification against others: class 0 for `waveform-40` and `splice`, class 1 for `satimage` and class 2 for `diabetes`. Low-fidelity labels were generated by flipping original labels with the specified probability (noise level).

3. Real datasets: we used `music_genre` (mscg) and `sentiment_polarity` (sntp) from [55], which had been annotated with crowd-sourcing. Each object in those datasets was labeled by multiple annotators, therefore we considered majority voting statistic over object labels as high-fidelity and a single random annotation as low-fidelity. Such an approach to model fidelity is reasonable in the context of crowd-sourced annotations where each of them costs some amount of resources (e.g. money or time). For example, some objects are easy to classify with machine learning algorithms, thus, one vote would be enough to annotate them, whereas for complex objects many votes are necessary for obtaining good confidence in labels. Finally, since `music_genre` dataset had multiple classes, we tested each of them with the one-vs-all scheme as separate datasets.

2.2.4.3 Comparison of methods

For datasets in groups 1 and 2 we compared the methods with the state-of-the-art hetmogp [24]. For datasets on crowd-sourcing annotation (group 3) we also compared our method with the state-of-the-art method gp-ma [55]. No comparison was made with the method of [56], since we couldn't find a publicly available source code.

[1] We published them in this repository `https://github.com/user525/mfgpc`



(A) dbts (B) stmg (C) mshr (D) wvfr

FIGURE 2.2: Comparison of predicted class probabilities with multi-fidelity MCMC and Laplace inference on datasets from group 2: typical cases of correlations.

TABLE 2.3: Comparison of ROC AUC in a single run for MCMC and Laplace inference on datasets from group 2 during verification tests. Margins indicate standard deviations of mean estimates.

dataset	MF gpc Laplace	MF gpc MCMC
dbts	0.834 ± 0.002	0.832 ± 0.002
grmn	0.755 ± 0.004	0.749 ± 0.005
hpth	0.636 ± 0.010	0.632 ± 0.015
mshr	0.999 ± 0.000	0.999 ± 0.000
stmg	0.997 ± 0.000	0.997 ± 0.000
spmb	0.933 ± 0.001	0.927 ± 0.002
splc	0.835 ± 0.054	0.836 ± 0.055
wvfr	0.655 ± 0.038	0.649 ± 0.038

At the outset, we verified our implementation of Laplace inference by comparing its predictions with those of MCMC (implemented with PyMC3 package [57]) with the same hyper-parameters on real datasets from group 2 ensuring that true posteriors are non-Gaussian. Each training set contained 75 randomly sampled high fidelity observations and flip probability of 0.2 in low-fidelity observations; we used 10 different random seeds to sample training sets for each dataset. The typical results of comparison are shown in Figure 2.2 and Table 2.3. The overall performance of two inference approaches is on par, whereas correlation behavior resembles the patterns observed in single-fidelity GPs classification ([5], figure 6), which lends evidence supporting the correctness of our method.

The main evaluation procedure was the following: for a single test, we selected a small random subsample of high fidelity observations and 3 times larger subsample of low fidelity observations. We trained all methods on those subsamples and evaluated predictions on the high-fidelity test set. For each dataset, we run 3 tests with different random subsamples, except `sentiment_polarity`, for which we run 15 tests, and `music_genre`, for which we run 30 tests (3 per class as noted in sec. 2.2.4.2).

We report average ROC AUC across all tests and methods in figures 2.3, 2.4, 2.5 (see also appendix with corresponding tables A.1, A.2, A.3) and table 2.4. For those tests,

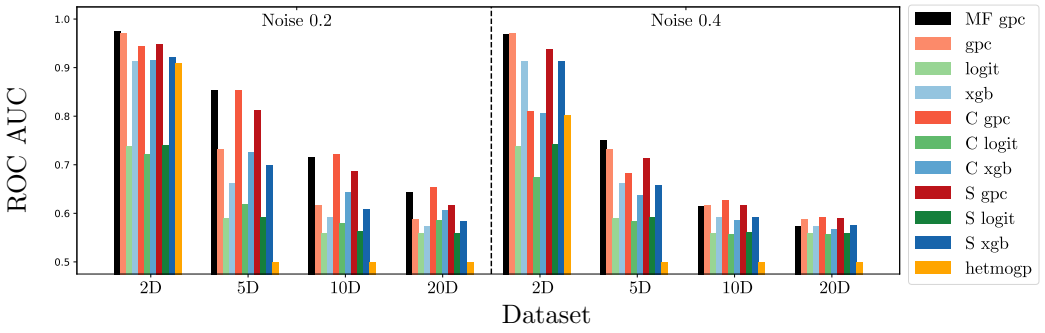


FIGURE 2.3: Average ROC AUC among multiple runs on artificial datasets from group 1.

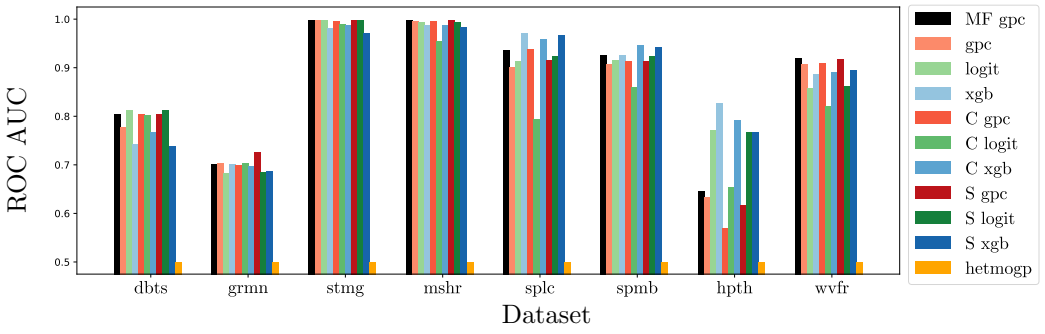


FIGURE 2.4: Average ROC AUC among multiple runs on datasets from group 2 with noise level 0.2.

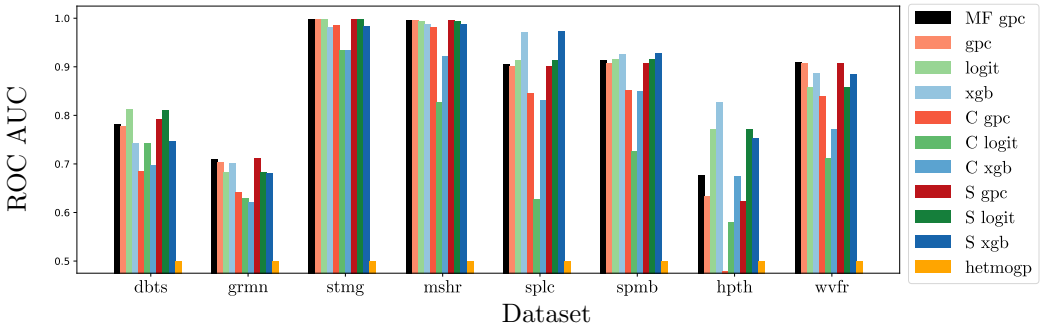


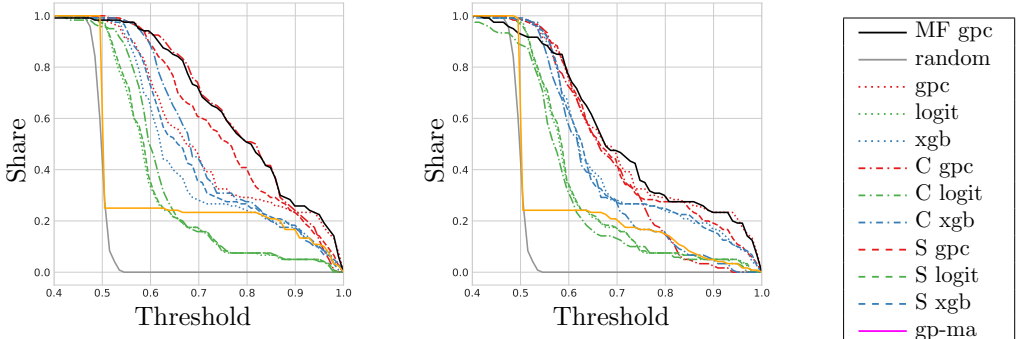
FIGURE 2.5: Average ROC AUC among multiple runs on datasets from group 2 with noise level 0.4.

each training set contained 75 high fidelity observations. The methods that performed not worse than 1 percent compared to the best result on the dataset are highlighted with bold.

Supplementary ROC AUC profiles are presented in figures 2.6 and 2.7. Overall, MF gpc has a good performance, except sntp dataset. Notably, on this dataset all GPs-based methods have poor performance, which is not surprising, since we used a translation-invariant isotropic kernel, which is not suited well for highly clustered non-stationary data.

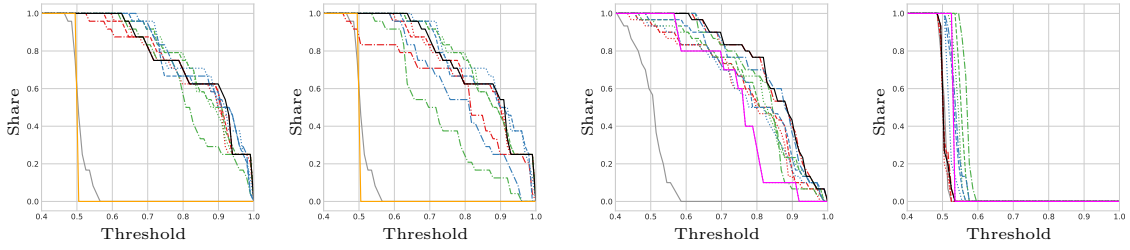
TABLE 2.4: Average ROC AUC among multiple runs on datasets from group 3 with natural noise. Margins indicate standard deviations of mean estimates.

method	mscg	sntp
MF gpc	$\mathbf{0.851 \pm 0.019}$	0.504 ± 0.003
gpc	0.772 ± 0.027	0.502 ± 0.002
logit	0.794 ± 0.025	0.542 ± 0.003
xgb	0.773 ± 0.026	0.520 ± 0.003
C gpc	$\mathbf{0.849 \pm 0.019}$	0.505 ± 0.003
C logit	0.812 ± 0.017	$\mathbf{0.569 \pm 0.003}$
C xgb	$\mathbf{0.843 \pm 0.020}$	0.538 ± 0.004
S gpc	0.785 ± 0.026	0.504 ± 0.002
S logit	0.797 ± 0.024	0.553 ± 0.003
S xgb	0.800 ± 0.024	0.533 ± 0.004
gp-ma	0.744 ± 0.018	0.531 ± 0.000



(A) Noise level 0.2 (B) Noise level 0.4

FIGURE 2.6: ROC AUC profiles for artificial datasets from group 1.

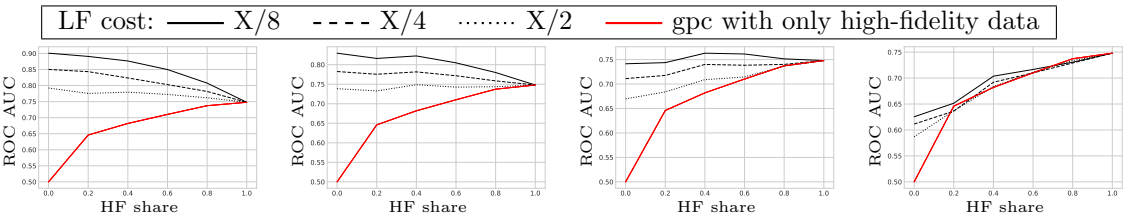


(A) Group 2 noise level 0.2 (B) Group 2 noise level 0.4 (C) Group 3 mscg (D) Group 3 sntp

FIGURE 2.7: ROC AUC profiles for real datasets from groups 2 and 3. Colors represent the same legend as in figure 2.6.

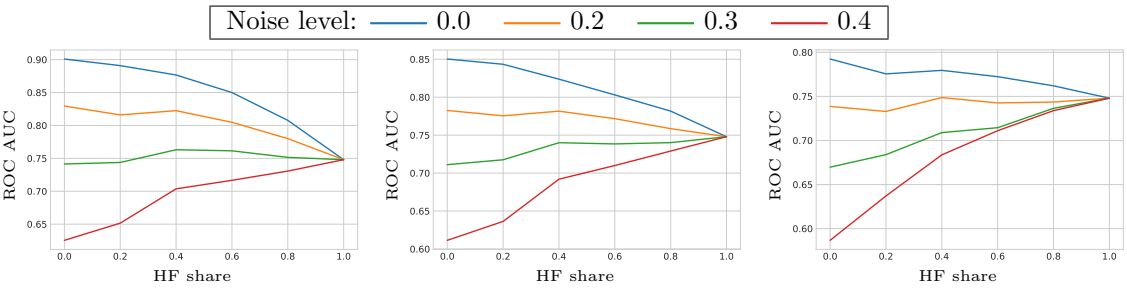
2.2.4.4 Budget distribution among variable fidelity sources

We studied how the ratio of low- and high-fidelity samples sizes affects the classification quality of MF gpc on datasets from group 1. An experimental setup was the following: we assumed each high-fidelity entry cost X units, whereas low-fidelity entries cost a fraction of X (with various fractions for different experiments). The training samples were formed based on the total budget: some part of it was allocated for high-fidelity



(A) Noise level 0.0 (B) Noise level 0.2 (C) Noise level 0.3 (D) Noise level 0.4

FIGURE 2.8: Performance of MF gpc depending on share of budget allocated to high-fidelity data (HF share) for different ratios of low-fidelity cost to high-fidelity cost.



(A) LF cost X/8 (B) LF cost X/4 (C) LF cost X/2

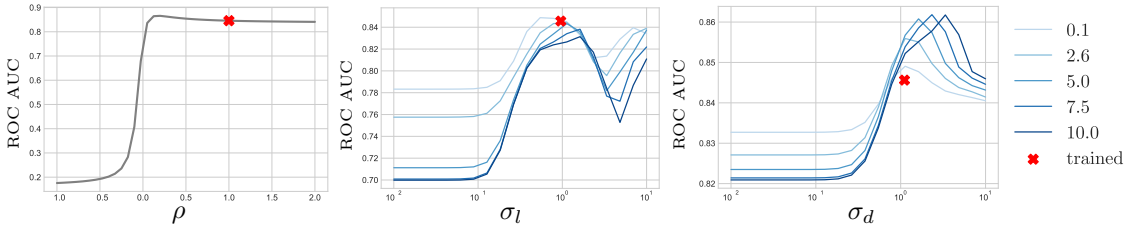
FIGURE 2.9: Performance of MF gpc depending on share of budget allocated to high-fidelity data (HF share) for different noise levels in low-fidelity labels.

data, the rest was for low-fidelity data. If the whole budget was spent on high-fidelity data, then the training sample contained 100 entries.

Figure 2.8 demonstrates the more low-fidelity data are available or the less noise is in it, the better classifier works w.r.t. the fixed amount of high-fidelity entries. That is, having fixed D_H, adding more data to D_L with the same noise level in low-fidelity labels does not reduce the quality of predictions of our classifier. In the worst-case scenario, when low-fidelity labels are independent of high-fidelity ones, for example they consist of merely random noise, MF gpc model degenerates to an ordinary gpc trained on D_H, because in co-kriging formula (2.3) component $\rho f_L(\mathbf{x}_i^H)$ becomes 0, thus, $f_H(\mathbf{x}_i^H) = \delta(\mathbf{x}_i^H)$.

Figure 2.9 shows that in case of low noise level in low-fidelity labels the sample size advantage overbalances the decreased labels quality, thus, spending all budget on low-fidelity data is the best option for this case. On the other hand, when the noise in low-fidelity is high, adding any amount of low-fidelity entries instead of high-fidelity ones to the training sample reduces the performance of the classifier.

These experiments show that in boundary cases single-fidelity gpc is the choice either for training on low-fidelity data when the noise in labels is low or for training on high-fidelity data when noise is high, whereas MF gpc works slightly better for intermediate noise levels in low-fidelity. It is not trivial to find the right balance in advance, but observations in this section can be used as a rule of thumb in practice.



(A) linear coefficient in co-kriging (B) $k_l(\cdot,\cdot)$ hyperparams (C) $k_d(\cdot,\cdot)$ hyperparams

FIGURE 2.10: Sensitivity of model performance to its hyperparameters in case of low or moderate noise in low-fidelity labels. Curves of different shades in figures 2.10b and 2.10c are associated with the the log-scale coefficient (s_* in (2.28)) of the corresponding kernel. Red mark indicates parameters and performance of the tuned model during the training.

2.2.4.5 Sensitivity to hyperparameters

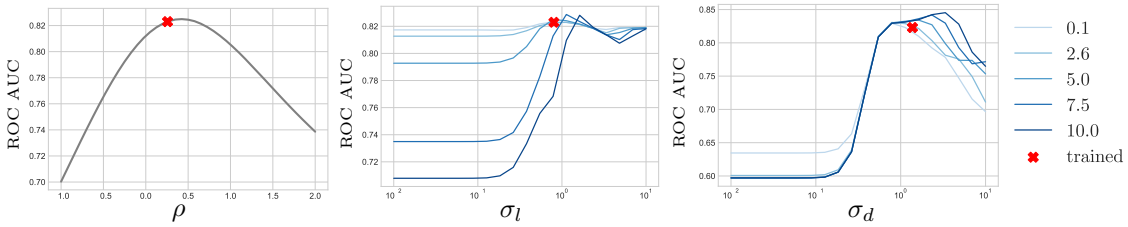
We used radial basis functions as kernels for Gaussian processes in the following form:

$$k_*(\mathbf{x}_i, \mathbf{x}_j) = exp(s_*)\ exp\left(-\frac{1}{2}\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma_*^2}\right), \tag{2.28}$$

where $(s_*, \sigma_*) = \boldsymbol{\theta}_*$ are kernel parameters, $* \in \{l, d\}$ indicates the corresponding Gaussian process.

In these series of experiments we first tuned the model on the training samples, then varied some hyperparameters while kept others fixed to their values obtained during the training. While ρ was varied, parameters of kernels were fixed. While $\boldsymbol{\theta}_l = (s_l, \sigma_l)$ was varied across the grid of s_l and σ_l values, parameters of k_d and ρ were fixed and vice versa for $\boldsymbol{\theta}_d = (s_d, \sigma_d)$. Eventually, for each combination of hyperparameters we estimated model's performance on the corresponding validation samples.

Figures 2.10 and 2.11 show a typical sensitivity of model's performance on the validation set with respect to the hyperparameters $\rho, \boldsymbol{\theta}_l, \boldsymbol{\theta}_d$ for cases with low or moderate noise and the case with high noise in low-fidelity data respectively. The former cases are characterized with low local sensitivity to ρ and a sharp decrease in performance when its sign changes; performance is also more sensitive to the parameters of k_l than to those of k_d. For latter cases the situation is opposite: the performance is more affected by local changes in ρ, regarding kernels the model is vice versa more sensitive to parameters of k_d than those of k_l.



(A) linear coefficient in co-kriging

(B) $k_l(\cdot,\cdot)$ hyperparams

(C) $k_d(\cdot,\cdot)$ hyperparams

FIGURE 2.11: Sensitivity of model performance to its hyperparameters in case of high noise in low-fidelity labels. Curves of different shades in figures 2.11b and 2.11c are associated with the the log-scale coefficient (s_* in (2.28)) of the corresponding kernel. Red mark indicates parameters and performance of the tuned model during the training.

2.3 Conclusions

Multi-fidelity modeling of discrete response surfaces can become useful in a number of applied disciplines, yet such methods have got little attention so far. In this work, we extended Laplace inference algorithm for classification based on GPs to make it work with multi-fidelity data. By modeling latent GPs dependency with a co-kriging schema, which has been used previously for multi-fidelity regression, our method can identify not only the overall relevance of low-fidelity data, but resolve local item-dependent discrepancies between fidelities due to inference on residual Gaussian process δ.

We evaluated our method on multiple artificial and real datasets with natural and various levels of simulated noise and compared its performance with a number of baseline approaches and state-of-the-art methods. We also experimentally studied under which conditions adding noisy low-fidelity to the training set increases quality on the top of high-fidelity data classification. Depending on the dataset nature, MF gpc can alternate its performance with respect to other methods, however, it is more resistant to different noise levels in low-fidelity labels. That is, when the classifiers based on GPs can learn datasets well, MF gpc has a top performance, whereas in other cases our method is on par with the considered methods.

Chapter 3

Multi-fidelity active search

Can a system discover what a user wants without the user explicitly issuing a query? A recommender system proposes items of potential interest based on past user history. On the other hand, active search incites and learns from the user feedback in order to recommend items that meet a user's current tacit interests, hence promises to offer up-to-date recommendations going beyond those of a recommender system. Yet extant active search methods require an overwhelming amount of user input, relying solely on such an input for each item they pick.

Consider the following scenario. A user visits an online bookstore looking for a new novel to buy. Given the large collection of options that the bookstore offers, it is very difficult to identify interesting titles. Keyword search provides a starting point, but still, the returned results may be too many to sift through. On the other hand, faceted search creates fixed views of the book catalog that do not zoom in on the books that would be of interest to the user. Similarly, lists of popular or highly-rated books offer general recommendations. These interfaces may serve as a good starting point but are far from helping the users quickly complete their task.

A way to meet the user's intent is to employ a recommender system [58]. However, the recommender systems require a high computational cost for retraining. Ideally, a system should interact with and learn from the user, seeking the user's feedback to assess its own guesses about the user's interests. That is the idea behind *active search* [13, 16, 59, 60]: an online learning mechanism that, given a set of items, a similarity measure between them, and previous user feedback, iteratively chooses the next item to present to the user for evaluation. This mechanism balances curiosity for unexplored items (exploration) with the need to meet the user's needs through this interaction (exploitation). The goal is to maximize a cumulative relevance function over all presented items until the user quits. Since the previous choices affect the following ones, active search is more

than a Markov Decision Process (MDP) [61]. The state-of-the-art active search method, GP-Select [16], models the user utility as a sample from a Gaussian Process and applies Gaussian Process Regression (GPR) to address the exploration–exploitation dilemma.

However, the conventional active search methods expect the user to provide the whole input they receive. This requirement imposes an overwhelming burden on the user. The question arises: Could we *combine* the user's feedback with insights extracted from other information sources in an online manner, so as to alleviate the burden on the user and deliver highly relevant results within a few interactions?

In this chapter, we propose an active search method that merges user inputs with those derived from other sources, as from a recommender system, so as to learn a continuous relevance score function that captures the user's interests. We treat the problem as one of regressions and apply the toolbox of Gaussian process regression with multiple target variables, i.e., co-kriging. The use of co-kriging is motivated by several applications in engineering and justified in terms of both conditional expectation and maximum-likelihood estimation [62, 63]. Our design belongs to the class of multi-fidelity methods [62, 64, 65], in which a low-fidelity function simulates expensive high-fidelity operations (e.g., car crash tests), so as to reduce the required amount of high-fidelity evaluations. Likewise, we bolster high-fidelity user evaluations by integrating them with correlated low-fidelity system-derived evaluations.

The low-fidelity function is learned contemporaneously with the high-fidelity function, also in active fashion. Our experiments with real and simulated user interactions show that this Multi-Fidelity Active Search with Co-kriging (MF-ASC) mechanism outperforms the state of the art under reasonably correlated fidelities.

3.1 Multi-fidelity active search and optimization methods

We survey the related work on active search and multi-fidelity optimization. Table 3.1 gathers together previous works' characteristics.

3.1.1 Active Search

Online similarity learning. *Online* methods learn by interacting with the user. MindReader [71] learns a distance function among the items in a database and the example items provided by the user, thereby inferring an implicit query expressed by weights over attributes. Other works follow a similar approach to *similarity learning* [72, 73].

	Active	Search	Regression	Multivalued	Bayesian	Multifidelity	Cokriging
Similarity learning	✘	✘	✔	✔	✘	✘	✘
SVM_{act} [66]	✔	✘	✘	✘	✘	✘	✘
BOAS [13]	✔	✔	✘	✘	✔	✘	✘
Soft-Label [59]	✔	✔	✘	✘	✘	✘	✘
GP-SOPT [60]	✔	✔	✔	✘	✔	✘	✘
GP-Select [16]	✔	✔	✔	✔	✔	✘	✘
MF-UCB [67]	✔	✘	✔	✔	✘	✔	✘
MISO [68]	✔	✘	✔	✔	✔	✔	✘
MF-PES [69]	✔	✘	✔	✔	✔	✔	✘
MF-GP-UCB [70]	✔	✘	✔	✔	✔	✔	✘
MF-ASC	✔	✔	✔	✔	✔	✔	✔

Table 3.1: Related work with present (✔) and absent (✘) affordances.

Such *online learning* methods adapt to user feedback, yet do not adaptively determine what feedback to ask for.

Classificatory Active Search. *Active learning* incrementally *selects* data items to learn from based on the previous observations [3]. *Active search* applies active learning to arrive at apt search results under a limited budget of the user feedback [59], balancing *exploration* of the user feedback on unknown values to improve its model and *exploitation* of that model to collect high-utility items. SVM_{act} [66] applies active learning to train a Support Vector Machine binary classifier. Bayesian Optimal Active Search (BOAS) [13] applies Bayesian decision theory to active search for the binary classification; Wang et al. [59] extend this idea to the graphs with a *soft-label* model by which the labeled nodes influence a query node in a manner diminishing by distance. Yet, such approaches collect the binary user feedback and predict utility by means of the binary classification [16].

Regressive Active Search. LinUCB [74] first proposed a ridge-regression technique with an upper confidence bound for recommendation by multi-armed bandits. GP-Select [16] models the user utility as a sample from a Gaussian Process and applies Gaussian Process Regression to address the active search exploration–exploitation dilemma. GP-SOPT [60] applies similar ideas on graphs, yet uses the binary user feedback values, even while predicting such values by regression. Thus, no previous work conducts active search on graphs using regression-based prediction and multi-valued reward values at the same time. Besides, these methods can only improve their predictions by accumulating the user feedback.

3.1.2 Multi-fidelity optimization

Multi-fidelity optimization is applied in the design of complex systems [64], where a computationally expensive high-fidelity objective function is approximated by a less expensive low-fidelity function and a few high-fidelity samples. For instance, in aeronautical design minimizing friction at supersonic speed, the high-fidelity function is a measurement on an aircraft wing, while the low-fidelity function is a computer simulation [75].

MF-UCB [67] first introduced an upper confidence bound for multi-fidelity function optimization by multi-armed bandits. MF-GP-UCB [70] improved this bound further applying predictions based on Gaussian Process Regression. Such works combine multiple inputs of diverse fidelities in order to achieve an optimization objective. However, they assume that those diverse inputs are samples from the same distribution, arising out of a single phenomenon. Thus, they disregard the potential different nature of such fidelities. By contrast, co-kriging treats multi-fidelity sources properly as samples from the diverse correlated phenomena. Besides, such function optimization methods are designed with an objective of function optimization rather than active search, i.e., they do not directly apply to a cumulative objective as required by active search. Similarly, MF-PES [69] performs function optimization by minimizing the predictive entropy, while posing restrictive assumptions on its objective function; thus, MF-PES cannot accommodate an active search objective either. Last, MISO [68] performs function optimization by combining multiple sources, yet it applies gradient-based optimization in order to calculate its acquisition criterion; thus, it is inappropriate for online active search applications where a gradient of the user interest cannot be derived conveniently. Overall, to our knowledge, no previous work conducts active search by combining multi-valued reward values via regression.

3.2 Problem statement

Active search is a process that progressively learns and meets the user's tacit interests. To do so, it seeks the user feedback on its own guesses, balancing exploration of the unknown with exploitation of the known. The user provides feedback by means of an undisclosed *user evaluation* function $w : \mathcal{X} \mapsto [0, 1]$ on any item $\mathbf{x}$ in a dataset $\mathcal{X}$. By our multi-fidelity design, the system also holds an internal approximation of the user relevance score, $\widetilde{w} : \mathcal{X} \mapsto [0, 1]$; we discuss $\widetilde{w}$ candidates in Section 3.4.

At each step the system retrieves an item $x \in \mathcal{X}$ from the dataset, obtains its score either from the user or internally from $\widetilde{w}$, and pays a *fixed operation cost*, $c \in \mathbb{R}^+$ for asking

the user and $\widetilde{c} \in \mathbb{R}^+$ for an internal evaluation, incurred, e.g., by a cloud computing service. The system may request the evaluation of the same item $x \in \mathcal{X}$ twice, once from the user and once internally; these two choices correspond to the *high-fidelity* (φ^H) and *low-fidelity* (φ^L) strategy, respectively. We call the set of items evaluated by the user so far $S^H \subseteq \mathcal{X}$, and the set of items evaluated by the system $S^L \subseteq \mathcal{X}$. The *utility* of a subset $S \subseteq \mathcal{X}$ is the total relevance of items in the S, $\mathcal{U}(S) = \sum_{x \in S} w(x)$. We aim to find a policy for selecting S^H and S^L that are expected to gain the highest utility by the end of the interaction, under a given budget Λ:

$$\underset{S^H S^L \subseteq \mathcal{X}}{\operatorname{argmax}} \quad \mathcal{U}(S^H) \quad \text{subject to} \quad c|S^H| + \widetilde{c}|S^L| \;\leq\; \Lambda$$

The utility involves only S^H since the user sees only S^H; internal low-fidelity evaluations of S^L are undisclosed to the user, to avoid any unconscious bias. Finding the optimal policy is computationally intractable even for binary w; some works approximate the optimal plan with one-step lookahead [76] or heuristic methods [77].

3.3 Multi-fidelity active search framework

Our framework of Multi-Fidelity Active Search with Co-kriging (MF-ASC) seeks and learns from the user feedback using two instruments: an *action engine* that implements a policy and an *approximate relevance function* $\widetilde{w}(\cdot)$ that estimates the user evaluation $w(\cdot)$.

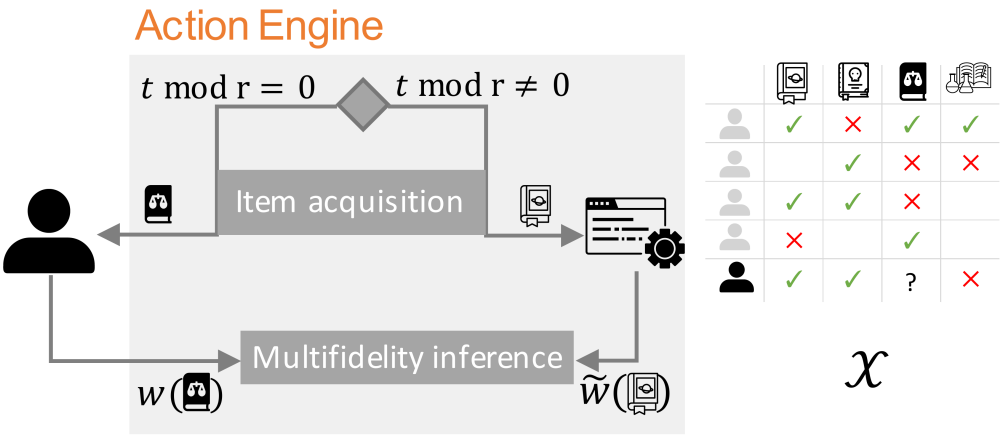


FIGURE 3.1: A step of the action engine at time t used for recommendation. The data $\mathcal{X}$ is a matrix of users and item ratings. MF-ASC selects one book to be rated, and the system either shows the item to the user for evaluation (when $t \mod r = 0$) or evaluates it internally with low fidelity.

3.3.1 The action engine

The action engine selects items for evaluation; it overcomes the problem's intractability and addresses the exploration-exploitation dilemma using the acquisition criterion of [78], also used in GP-Select [16]. Our contribution with respect to [16] is that we provide the means to integrate high- and low-fidelity sources by Bayesian multi-fidelity inference. A parameter $r \geq 0$ determines the ratio of low-fidelity to high-fidelity calls; this parameter can be fixed in advance or decided by the user. The problem of adapting r to given needs is orthogonal to our work.

Algorithm 2 presents our action engine. For each fidelity choice, it first computes the parameters of a regression model using the current state information (Lines 2 and 6); it returns the item $x \in \mathcal{X}$ that maximizes an Upper Confidence Bound (UCB)[1] acquisition criterion [78] by the regression model for either fidelity, excluding previously chosen items (Lines 3, 8, 10). The β_t parameter balances exploration and exploitation: large β_t favors exploring items having high $\sigma(x)$, i.e., *uncertainty* about their relevance, while small β_t favors exploiting items having high $\mu(x)$, i.e., *value* of relevance.

We emphasize that our system design is independent of the choice of acquisition criterion. A choice better than UCB would improve performance while leaving the multi-fidelity inference mechanism intact. Yet UCB yields strong regret guarantees, as we explain in Section 3.3.3.

[1] For $t = 0$, when no evaluation has been already provided, it returns a random item.

Algorithm 2 Action engine

Input: Data $\mathcal{X}$, S^H, S^L, k_0, Time t

Params: Fidelity ratio r

if $t \mod r = 0$ **then**

// Where ρ is obtained by eq. (3.7), $\mu(x) \leftarrow \mu^w(x)$; $\sigma(x) \leftarrow k^w(x,x)$

$(\rho, \mu, \sigma) \leftarrow$ MF-INFER-HIGH(S^H, S^L, k_0)

// Acquisition criterion in [78]

$x_t \leftarrow \underset{x \in \mathcal{X} \setminus S^H}{\operatorname{argmax}}\ \mu(x) + \beta_t \sigma(x)$

$S^H \leftarrow S^H \cup \{x_t\}$

else

// Where $\mu(x) \leftarrow \mu^{\widetilde{w}}(x)$; $\sigma(x) \leftarrow k^{\widetilde{w}}(x,x)$;

$(\mu, \sigma) \leftarrow$ MF-INFER-LOW(S^H, S^L, k_0)

// If negative correlation return lower-confidence bound

if $\rho > 0$ **then**

$x_t \leftarrow \underset{x \in \mathcal{X} \setminus S^L}{\operatorname{argmax}}\ \mu(x) + \beta_t \sigma(x)$

else

$x_t \leftarrow \underset{x \in \mathcal{X} \setminus S^L}{\operatorname{argmin}}\ \mu(x) - \beta_t \sigma(x)$

end if

$S^L \leftarrow S^L \cup \{x_t\}$

end if

3.3.2 Multi-fidelity inference

We now discuss the details of our inference model. To estimate the value and uncertainty of the relevance score at each node, with either fidelity, we assume that score at time t is a random variable sampled from a Gaussian distribution, and apply *co-kriging* [62] (i.e., Gaussian process regression over multiple target variables) informed by the data structure and previous scores.

Prior Distributions

We assume that *similar items* are more likely to get the same score. Then, the *prior* joint probability distribution of the relevance score function w is a Gaussian $\mathcal{N}(0, \mathbf{K})$ with mean 0 and a covariance matrix $\mathbf{K}$ reflecting item similarity. To capture this similarity, we face a choice between (i) *implicit* parameterization, which constructs a covariance function *directly* from the item matrix; and (ii) *explicit* parameterization $p : \mathcal{X} \to \mathbb{R}^m$, which first embeds items in a low-dimensional vector space and then applies a parametric covariance function (kernel) k on each pair of items $x_i, x_j \in \mathcal{X}$ — typically a squared exponential kernel $k(x_i, x_j) = \eta e^{-\theta ||p(x_i) - p(x_j)||^2}$ — to compute a covariance matrix $\mathbf{K} = [k(x_i, x_j)]_{i,j=1}^{|\mathcal{X}|}$, while estimating kernel parameters $\{\eta, \theta\}$ by maximum likelihood estimation [62]. Explicit parameterization is flexible due to its

tunable parameters. On the other hand, it is complicated and incurs information loss as a result of embedding.

We opt for a compromise between the explicit way (which considers relations between all item pairs) and the implicit way (which considers only adjacency connections): first, we compute a similarity matrix $\mathbf{S}$, that contains pairwise similarities among all items in the dataset $\mathcal{X}$ (examples are discussed in Section 3.4). Then we extract the Laplacian of $\mathbf{S}$, $\mathbf{L} = \mathbf{D} - \mathbf{S}$, where $\mathbf{D}$ is a diagonal matrix obtained by summing the values in each row of $\mathbf{S}$, $\mathbf{D}_{ii} = \sum_{j=1}^{|\mathcal{X}|} \mathbf{S}_{ij}$. Last, we compute a covariance matrix from the Laplacian of $\mathbf{S}$, $\mathbf{K}_0 = (\mathbf{L} + \lambda \mathbf{I})^{-1}$, where λ is a regularization parameter that determines how much credit is to be given to $\mathbf{S}$. Henceforth, we denote the corresponding kernel function defined on pairs of the dataset as k_0.

Posterior Distributions

We now obtain the means μ and standard deviations σ of the *posterior* probability distributions for relevance score random variables at time t, based on Gaussian priors and the sets of items S^H and S^L evaluated at time t. We represent the user and system scores for all items S^H and S^L evaluated by time t as a vector $\mathbf{y} = (\mathbf{y}^L, \mathbf{y}^H)^\top$, where vector $\mathbf{y}^L$ holds low-fidelity evaluations and vector $\mathbf{y}^H$ holds high-fidelity ones.

Low-Fidelity Inference (MF-Infer-Low). Let t^L be the number of items evaluated by low fidelity at time t; assuming without loss of generality that the order of items keeps the evaluated ones before the rest, we denote:

$$\mathbf{k}^{\widetilde{w}}(x) = [k_0(x_1, x), ..., k_0(x_{t^L}, x)]^\top \tag{3.1}$$

$$\mathbf{K}_0^{\widetilde{w}} = \left[k_0(x_i^L, x_j^L)\right]_{i,j=1}^{t^L} \tag{3.2}$$

$\mathbf{K}_0^{\widetilde{w}}$ is a square matrix made out of the contents of the covariance matrix $\mathbf{K}_0$ for items already evaluated by low fidelity at time t. By Bayesian inference calculations [63], the parameters of the *low-fidelity* posterior distribution $\sim \mathcal{N}(\boldsymbol{\mu}^{\widetilde{w}}, \mathbf{K}^{\widetilde{w}})$, are

$$\boldsymbol{\mu}^{\widetilde{w}} = \left[\mu^{\widetilde{w}}(x_i)\right]_{i=1}^{|\mathcal{X}|}; \qquad \mu^{\widetilde{w}}(x) = (\mathbf{k}^{\widetilde{w}}(x))^\top (\mathbf{K}_0^{\widetilde{w}} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{y}^L \tag{3.3}$$

$$\mathbf{K}^{\widetilde{w}} = \left[k^{\widetilde{w}}(x_i, x_j)\right]_{i,j=1}^{|\mathcal{X}|}; \; k^{\widetilde{w}}(x_i, x_j) = k_0(x_i, x_j) - \mathbf{k}^{\widetilde{w}}(x_i)^\top (\mathbf{K}_0^{\widetilde{w}} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{k}^{\widetilde{w}}(x_j) \tag{3.4}$$

where σ_n^2 denotes the variance of noise associated with the model. We give Equation 3.4 in its general form, yet calculate only the diagonal entries $k^{\widetilde{w}}(x, x)$ of this covariance matrix, as we only need the variance values in single items (Algorithm 2, Lines 2 and 6).

High-Fidelity Inference (MF-Infer-High). Next, we assume that the high-fidelity (user) score function $w(\cdot)$ is correlated with the low-fidelity (system) score function $\widetilde{w}(\cdot)$,

and exploit this correlation for inferring high-fidelity. The user score should then be a linear combination of two independent stationary Gaussian processes over the items $\mathcal{X}$, namely $\widetilde{w}(\cdot)$ scaled by a factor ρ and a Gaussian process δ capturing the said correlation; thus, for an item $x \in \mathcal{X}$:

$$w(x) = \rho \cdot \widetilde{w}(x) + \delta(x). \tag{3.5}$$

As noted in the paragraph about the prior distributions, the priors of $\widetilde{w}$ and δ when the interaction starts, based on item similarities, are Gaussians. Let f denote either $\widetilde{w}$ or δ. Then $\boldsymbol{f} = (f(x_1), ..., f(x_{|\mathcal{X}|})) \sim \mathcal{N}(0, \mathbf{K}_0)$, thus,

$$\Pr\{\boldsymbol{f}{=}\boldsymbol{z}\} \propto \exp\left(-\tfrac{1}{2}\left(\textstyle\sum_{i,j=1}^{|\mathcal{X}|} \mathbf{S}_{ij}(z_i^2 - z_i z_j) + \lambda \sum_{j=1}^{|\mathcal{X}|} z_j^2\right)\right), \tag{3.6}$$

where $\boldsymbol{z} = (z_1, ..., z_{|\mathcal{X}|}) \in \mathbb{R}^{|\mathcal{X}|}$, λ is the regularization parameter in the computation of $\mathbf{K}_0$ from the Laplacian of $\mathbf{S}$. The co-kriging correlation (Eq. 3.5) allows us to build the inference of high-fidelity user relevance on the top of the low-fidelity one. We calculate ρ as:

$$\rho = \frac{(\mathbf{y}^H)^\top \mathbf{K}_0^\delta \, \boldsymbol{\mu}^{\widetilde{w}}(S^H)}{(\boldsymbol{\mu}^{\widetilde{w}}(S^H))^\top \mathbf{K}_0^\delta \, \boldsymbol{\mu}^{\widetilde{w}}(S^H)}, \tag{3.7}$$

where $\boldsymbol{\mu}^{\widetilde{w}}(S^H) = [\mu^{\widetilde{w}}(x_1^H), ..., \mu^{\widetilde{w}}(x_{t^H}^H)]^\top$ and $\mathbf{K}_0^\delta$ is made out of the contents of the covariance matrix for items already evaluated by high fidelity at time t, $[k_0(x_i^H, x_j^H)]_{i,j=1}^{t^H}$. Eventually, we obtain high-fidelity posteriors, $\mu^w(x)$ and $k^w(x_i, x_j)$, as in equations (3.3) and (3.4), using the vector $[\mathbf{y}^L, \mathbf{y}^H]$ obtained concatenating vector $\mathbf{y}^L$ and $\mathbf{y}^H$ in place of $\mathbf{y}^L$ and the following combined covariances in place of $\mathbf{k}^{\widetilde{w}}$ and $\mathbf{K}_0^{\widetilde{w}}$, respectively:

$$\mathbf{k}^w(x) = \begin{pmatrix} \rho \mathbf{K}_0^{\widetilde{w}}(S^L, \{x\}) \\ \rho^2 \mathbf{K}_0^{\widetilde{w}}(S^L, \{x\}) + \mathbf{K}_0^\delta(S^H, \{x\}) \end{pmatrix} \tag{3.8}$$

$$\mathbf{K}_0^w = \begin{pmatrix} \mathbf{K}_0^{\widetilde{w}}(S^L, S^L) & \rho \mathbf{K}_0^{\widetilde{w}}(S^L, S^H) \\ \rho \mathbf{K}_0^{\widetilde{w}}(S^H, S^L) & \rho^2 \mathbf{K}_0^{\widetilde{w}}(S^H, S^H) + \mathbf{K}_0^\delta(S^H, S^H) \end{pmatrix} \tag{3.9}$$

where $\mathbf{K}(A, B)$ denotes the matrix $[k(a_i, b_j)]_{i,j}$ of pairwise correlations between items in sets A and B. These results conclude our discussion of multi-fidelity inference (i.e., Lines 2 and 6 of Algorithm 2). We reiterate the fact that, low-fidelity calculations are also appropriately used in high-fidelity inference.

3.3.3 Algorithm Complexity and Regret Guarantees

The complexity of MF-ASC depends on the outlined inference method, which computes the posterior at each iteration of Algorithm 2 and applies it to eligible nodes (Line 3). This computation takes $\mathcal{O}(|\mathcal{X}_t|^3 + |\mathcal{X}| \cdot |\mathcal{X}_t|^2)$, where $\mathcal{X}_t = S^H \cup S^L$ is the set of low- and high- fidelity evaluations performed by time t, $|\mathcal{X}_t|^3$ stands for matrix inversion in

equations 3.3 and 3.4, and $|\mathcal{X}| \cdot |\mathcal{X}_t|^2$ stands for matrix-vector multiplications for each $x \in \mathcal{X}$. We reiterate that, as we need to know only the variances in items, we only need to calculate the posterior covariances along the diagonal. Hence, the overall complexity of MF-ASC per iteration is $\mathcal{O}(|\mathcal{X}_t|^3 + |\mathcal{X}| \cdot |\mathcal{X}_t|^2)$; that is *cubic only* in the number of evaluations, which is constrained by the budget Λ; the complexity is *linear* in the number of nodes.

Since the datasets are finite discrete structures, the regret bounds in [78] are attributable to our method. Thus, if we set the exploration-exploitation tradeoff parameter to $\beta_t = \sqrt{2\log(|\mathcal{X}|t^2\pi^2/6\delta)}$, then our action engine has a regret guarantee $\mathcal{O}(\sqrt{T\gamma_T \log|\mathcal{X}|})$ with probability $1-\delta$, where T is the number of user interactions, $\gamma_T = \frac{1}{2}\sum_{t=1}^{T} \log(1 + \sigma_n^{-2}\sigma_{t-1}(x_t))$, and σ_{t-1} the posterior variance at time $t-1$ (Algorithm 2, Line 2).

3.4 Applications of MF-ASC

MF-ASC facilitates effective active search on different data types. An exhaustive study of data types to use our framework on is outside the scope of this work. Here, we suggest two practical domains: *consumer recommendation* and *information graph exploration*.

The framework has two key components: a *similarity function* defined between pairs of objects and *fidelity functions*. The low-fidelity function, in particular, is critical for the quality of predictions. An inadequate choice would let the posterior estimates degenerate to a random predictor [17], due to the correlation by co-kriging (Equation (3.5)). An effective low-fidelity function may be based on historical user interactions, such as query logs and domain knowledge. The design of such an elaborate fidelity function falls outside the scope of this work. We present a general framework and instantiate it with uncomplicated functions to highlight the benefits of the framework as such.

3.4.1 Case 1: Consumer Recommendation

We first apply our active search framework in the domain of a classical consumer-rating-based recommender system. A conventional recommender system estimates user preferences based on the previous interactions only. The data $\mathcal{X}$ is a matrix of users and items, where each user expresses preferences on one or more items. We show the effect of upgrading such a system with our MF-ASC mechanism: the system itself provides a low-fidelity input, while it also invites the user to score the selected items, so as to progressively learn the user's current preferences and improve its recommendations.

Similarity function. In general, the similarity between items can be obtained from their attributes. The particular form of similarity depends on the downstream application. In collaborative filtering recommendations, one can obtain similarity from the distance between items' latent factors, constructed, e.g., by Singular Value Decomposition (SVD) of the user-item preference matrix [79].

Fidelity functions. In this case, we take the predictions of a conventional recommender algorithm as the low-fidelity function and real user's input as the high-fidelity function.

3.4.2 Case 2: Information Graph Exploration

Next, we expand our study to the domain of information graphs. An *information graph* is a quadruple $G : \langle V, E, \phi, \psi \rangle$, where V is a set of nodes, $E \subseteq V \times V$ is a set of edges, $\phi : V \mapsto L_V$, $\psi : E \mapsto L_E$ are node and edge labeling functions, respectively. Information graphs present a particularly hard domain for learning algorithms due to their high dimensionality and non-trivial structural relationships. We show the effect of applying MF-ASC for search over such a structure; in the absence of rating data, we simulate the low-fidelity input that a recommender system would provide as a corrupted version of the user's high-fidelity input.

Similarity function. We compute a node similarity matrix $\mathbf{S}$ (Section 3.3.2) in a semi-supervised manner: We add an edge between nodes with probability proportional to the Jaccard similarity of their textual labels; in case of multi-attribute nodes, we take the weighted sum of the Jaccard similarities among attributes of the same type. Then we extract a node2vec [80] 50-dimensional embedding, using 50 random walks of length 10 from each vertex, window size 5, and a skip-gram model with negative sampling 5. Last, cosine similarities of node embeddings give the entries of $\mathbf{S}$.

Fidelity functions. We represent a user's intentions via an *intended set* $\mathcal{I}$, i.e., a set of nodes that the user tacitly wants to find. We construct $\mathcal{I}$ as the result of a query, which the user is presumably unable to formulate explicitly, and simulate the *high-fidelity* score w for a vertex v as the average cosine similarity from v to nodes in $\mathcal{I}$, normalizing all scores to [0, 1].

A *low-fidelity* function should approximate the preferences of a user. We derive such functions from high-fidelity ones by introducing controlled errors, using a k-nearest-neighbor (kNN) approach: for a given node v, we remove ℓ nodes from the set of its k NNs, and calculate low-fidelity values as the average of high-fidelity values for non-removed neighbors; by tuning $\ell \leq k$ and k, we obtain low-fidelity functions with an arbitrary correlation to high fidelity.

3.5 Experiments

3.5.1 Experimental methodology

We experiment with real and simulated users, on both consumer recommendation and information graph exploration. We measure the **quality** of obtained output in terms of *relative regret*. The *regret* is the deviation of utility from the optimal utility. We compute the optimal utility $\mathcal{U}^*_\Lambda$ for a given budget Λ by summing the relevance values w of the ground-truth top-Λ nodes; then, the regret is $\mathcal{U}^*_\Lambda - \mathcal{U}_\Lambda$, where $\mathcal{U}_\Lambda$ is the sum of utilities attained by the evaluated method by the end of the interaction. Then the *relative regret* is the ratio between the obtained regret and the average regret of the random method RAND over 50 runs: $\frac{\mathcal{U}^*_\Lambda - \mathcal{U}_\Lambda}{\mathcal{U}^*_\Lambda - \mathcal{U}^{\text{RAND}}_\Lambda}$. In each experiment, we report the average over examined instances; occasionally, we refer to this *average relative regret* simply as *Regret*; dashed curves show the 0.2- and 0.8-quantiles for corresponding curves of the same color. Relative regret measures performance with respect to a random acquisition criterion; the line Regret=1 represents the performance of RAND. In Section 3.5.6 we also report Recall@Λ, i.e., the ratio of relevant items (i.e., items in the intended-set) in the top-Λ elements returned by each method. We also report the **time** needed to select the next item to evaluate and analyze the scalability of the methods in Section 3.5.9.

Datasets. We experiment on three datasets with real user data:

- YAHOO: The Yahoo music dataset from the KDD-Cup 2001 [81], containing song ratings on a scale from 0 to 100. We extracted 5% of the most active users and the most rated songs, obtaining 50k users and 31k songs.
- YELP: A dataset from YELP challenge[2] containing business ratings on a scale from 1 to 5. We selected users and businesses that have at least 100 reviews, obtaining 68k users and 12k businesses.
- ACL[3]: A graph expressing the research interests of 28 researchers across 597 papers published in the Association for Computational Linguistics conference from 2000 to 2006, presented as lists of paper IDs; we used these lists as *intended sets*. We extract similarities among papers using the full graph to which they belong, including papers from other venues, and use the papers' term-frequency feature vectors for cosine similarity calculations (see Section 3.5.3).

In addition, we have devised *simulated* user data with these real-world graph data:

[2] https://www.yelp.com/dataset_challenge
[3] http://www.comp.nus.edu.sg/ sugiyama/SchPaperRecData.html, 'dataset 1'

• FREEBASE: We downloaded a snapshot of the Freebase[4] knowledge graph and extracted a *computer* domain, FB-Comp, containing information about computer models, manufacturers, software, and hardware, as well as a *fiction* domain, FB-Fict, containing information on novels.

• MICROSOFT ACADEMIC GRAPH: We extracted two samples from a network[5] of authors, publications, affiliations, and venues: MAG is a subgraph related to Computer Science conferences, with a taxonomy of topics related to keywords; MAG-Sm is a subset of MAG containing only papers. We calculate similarities among nodes in MAG-Sm using edges in MAG as well.

The table below lists the characteristics of graph datasets: number of edges $|E|$, vertices $|\mathcal{X}|$, node labels $|L_V|$, edge labels $|L_E|$; *avg*, *min*, and *max* node degree; modularity [82]; and density, $|E|/\binom{|\mathcal{X}|}{2}$.

	Size				*Degree*		
Dataset	$\|V\|$	$\|E\|$	$\|L_V\|$	$\|L_E\|$	Avg/Min/Max	Mod.	Density
FB-Comp	9.7k	18k	9.7k	70	3.7/1/1082	0.70	4×10^{-4}
FB-Fict	2.5k	16k	2.5k	74	13.2/6/1081	0.63	5×10^{-3}
MAG	6k	14k	6k	5	4.8/1/391	0.69	8×10^{-4}
MAG-Sm	1.1k	3.8k	1.1k	1	-/-/-	-	-
ACL	597	614	597	1	-/-/-	-	-

User simulation. On graph data with simulated users, we construct simulated user feedback as a preference score w arising from a structured query on the graph, as discussed in Section 3.4.2. We sample 5 nodes uniformly at random from the set of query results and use them as the initial set with all methods. We hand-crafted 25 queries for FB-Fict, 17 for FB-Comp, 19 for MAG, and 43 for MAG-Sm. All constructed queries for the first three datasets return at least 15 results with no or little overlap. The table below shows examples of such queries.

Dataset	*Query description*
FB-Comp	Programming languages influenced by Python.
	Soft with open source license.
FB-Fict	Fiction characters who have a super ability to fly.
	Fiction characters who are parents of students.
MAG	Authors that published papers alone.
	Affiliations in Germany related to Genomics field.
MAG-Sm	All papers from field Anomaly Detection.
	All papers from field User Modeling.

3.5.2 Implemented algorithms

We implemented a baseline method, a degenerate (randomized) active search variant, two single-fidelity active search algorithms derived from previous works, and a multi-fidelity method designed for function optimization adjusted to active search purposes. In particular, we compare MF-ASC against the following contestants:

[4]https://developers.google.com/freebase/data

[5]https://www.microsoft.com/en-us/research/project/microsoft-academic-graph/

• LabelProp: a baseline method that applies label propagation [83] with a k-nearest-neighbor kernel at each iteration and returns the node having the maximum probability to belong to Class 1 for user evaluation. We embed the similarity matrix $\mathbf{S}$ in a 2-dimensional[6] space using t-SNE [84] and run the implementation of label propagation in scikit-learn[7]. As the algorithm is designed for classification, we convert user evaluations provided at previous iterations to binary, marking a node as positive (class 1) if its user relevance score deviates from the maximum by at most 0.1.

• Rand: a method with an acquisition criterion that randomly selects a node to be evaluated.

• GP-SOPT: the single-fidelity active search method in [60] that uses the Laplacian of the data set's similarity matrix $\mathbf{S}$ and a σ-optimality criterion. We include this method in our study for the purpose of establishing the superiority of the UCB acquisition criterion of the σ-optimality criterion. We do that by constructing the following single-fidelity active search method, which represents the best of previous work.

• GP_Lapl: a single-fidelity active search method that combines the best choices of previous work: applies the UCB acquisition criterion, as GP-Select [16], and represents the data set by the Laplacian of its similarity matrix $\mathbf{S}$, as GP-SOPT [60]. We set $\beta_i = 0.01$, as explained in the paragraph below. The default form of GP_Lapl operates with high-fidelity input only, i.e., with user input as an active search method. We also create a variant, LF-only, that works with low-fidelity input only, ergo like a conventional recommender system that does not request an online user input.

• MF-GP-UCB: the state-of-the-art multi-fidelity function optimization method [70]. We adjusted the algorithm to an active search setting by disallowing the querying of the same item and fidelity pair more than once.

In order to make a fair comparison, we tune GP_Lapl on the best value of the exploration-exploitation tradeoff β_t. Recall that a large β_t favors exploration, skewing the selection towards nodes with high uncertainty. Figure 3.2 reports the results on the largest (FB-Comp) and the smallest (MAG-Sm) dataset; we witnessed similar performance for the other datasets. We note that $\beta_t \sim 0.01$ is the optimal choice for GP_Lapl; thus, we set $\beta_t = 0.01$.

[6] We tried higher dimensionality, without much improvement, hence we sticked to 2.

[7] http://scikit-learn.org/stable/modules/label_propagation.html

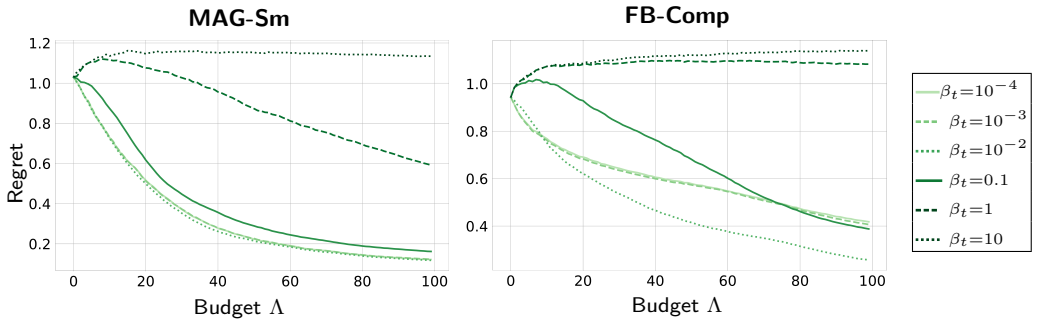


FIGURE 3.2: Regret of GP_LAPL vs. Λ and β_t.

Parameter settings. We set $c = 1$ and $\widetilde{c} = 0$, hence the budget Λ expresses the number of user interactions. We set β_t to 0.001 for high-fidelity and 0.01 for low-fidelity, steering low-fidelity evaluations towards exploration and high-fidelity ones towards exploitation; other values yielded worse results. We set the regularization parameter λ to 0.01, as in [60]. Unless otherwise indicated, we set the fidelity ratio in Algorithm 2 to $r = 5$, corresponding to one round of user feedback for every 5 low-fidelity evaluations. We assume a default correlation of 1 between the two fidelities. We delve into the effect of different correlation values in Section 3.5.7. Last, we deem the computation of the similarity matrix $\mathbf{S}$ in Section 3.4 as preprocessing for all algorithms.

The table below provides the default parameter values used in the experiments unless otherwise stated.

Parameter	*Default value*	*Meaning*
c	1	User evaluation cost
$\widetilde{c}$	0	System evaluation cost
β_t	high-fid: 0.001;low-fid: 0.01	Exploration/exploitation tradeoff (Alg. 2)
r	5	Low-high fidelity ratio (Alg. 2)
λ	0.01	Regularization parameter for $\mathbf{K}_0$

Summary of results. Our results with real-user data, reported in Section 3.5.3, confirm the advantage gained over the single-fidelity method, GP_LAPL, as well as over the multi-fidelity method MF-GP-UCB by using co-kriging for fidelity fusion. Section 3.5.4 investigates our choice of the acquisition criterion, showing that the GP_LAPL acquisition criterion we employ is preferable to that of GP-SOPT. In Section 3.5.5 we study the effect of the fidelity ratio r. Section 3.5.6 establishes the superior performance of MF-ASC over state-of-the-art methods in an ideal case, while Section 3.5.7 shows that MF-ASC can predict simulated users up to 3× faster if the correlation between low- and high- fidelity is at least 0.5. Section 3.5.8 illustrates that MF-ASC outperforms single-fidelity methods even when the simulated user input is discretized. Last, Section 3.5.9 shows that MF-ASC achieves real-time performance with a fast learning rate.

3.5.3 Real User Preferences

First, we test algorithms on preferences derived from real users on the domains discussed in Section 3.4: consumer recommendation and information graph exploration.

Consumer Recommendation. In this experiment, we use the Yelp and Yahoo data. High-fidelity evaluations are provided by recorded user-item ratings. Since these ratings are incomplete, we restrict the search area of the high-fidelity function during tests for all algorithms, so that they only query high-fidelity for items on which there is a recorded user score. Thus, we use the score that the real user in question would provide. On the Yelp data, we define the low-fidelity function as the average score of an item (i.e., business) by all friends of a user, if any. We obtain a low-fidelity function on the Yahoo data by partitioning it into *training* (60%) and *testing* (40%) parts along the time dimension. Testing data provide high-fidelity evaluations, while training data provide low-fidelity predictions on the testing data by means of collaborative filtering with SVD.

We test on 30 randomly selected users for each dataset. Here, we set $c = \frac{5}{6}$ and $\widetilde{c} = \frac{1}{6}$ in MF-GP-UCB, which balance low-fidelity calculations in the same manner as $r = 5$ does in MF-ASC. Figure 3.3 presents our cumulative results. On both data, MF-ASC outperforms LabelProp and MF-GP-UCB, as well as the random baseline by a wide confidence margin. On the Yahoo data, the single-fidelity method, GP_Lapl, performs better than multi-fidelity methods. This is due to weak correlations between low and high fidelity: the average Pearson coefficient between fidelities on the Yahoo data is 0.13, whereas on the Yelp data it is 0.4. Therefore, on the Yahoo data, multi-fidelity methods cannot regain the budget spent on exploring the dependence between fidelities. Despite this poor performance of multi-fidelity methods on the Yahoo data, MF-ASC clearly outperforms MF-GP-UCB. This result testifies the virtues of the co-kriging approach to data source fusion.

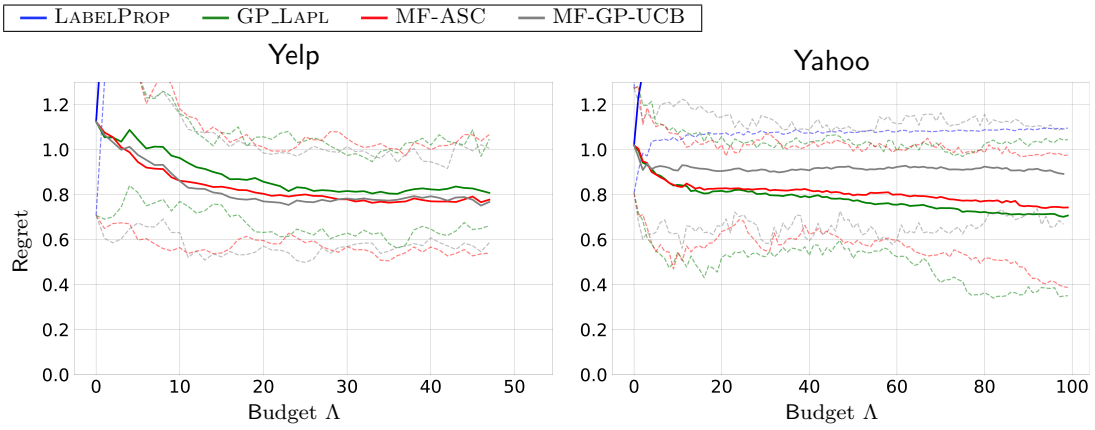


Figure 3.3: Relative regret vs. Λ on real datasets. Solid curves show the average and dashed curves show the 0.2- and 0.8-quantiles.

In the above, we examined a single-fidelity method that uses only the high-fidelity component of MF-ASC. In order to complete our study, we need to also examine a single-fidelity method that uses only the low-fidelity component of MF-ASC, hence behaves like a traditional recommender system. Figure 3.4 presents our results with such a method, LF-ONLY. We observe that LF-ONLY fares much worse than the active search methods, reaffirming the advantage of active search over a conventional recommender system.

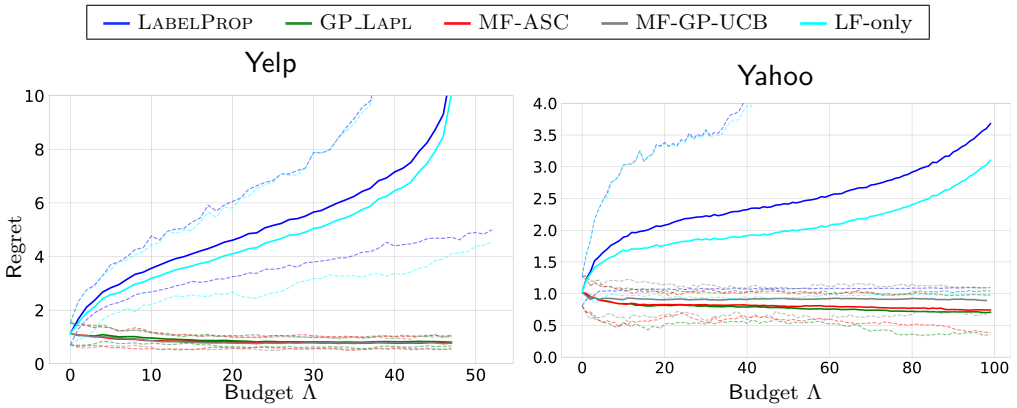
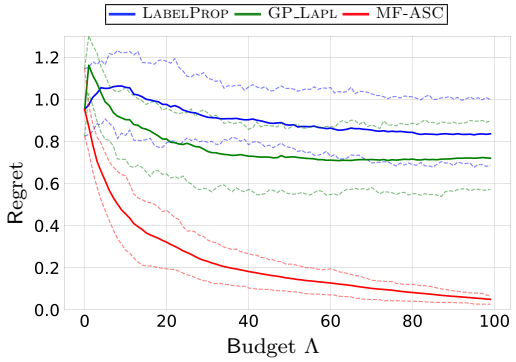
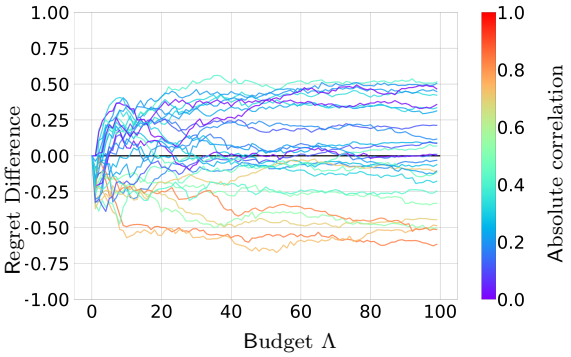


FIGURE 3.4: Relative regret vs. Λ on real data with LF-only. Solid curves show the average and dashed curves show the 0.2- and 0.8-quantiles.

Information Graph Exploration. Here, we use the ACL dataset. We obtain high-fidelity evaluations by averaging the term-frequency-vector cosine similarity between a paper and the papers in a researcher's *intended set*. Figure 3.5a shows the results for a best-case scenario, where low-fidelity has correlation=1.0 with high-fidelity. In a more realistic scenario, low-fidelity evaluations are derived by cosine similarity between a paper and the researcher's most recent papers (not the full list). Figure 3.5b shows the regret difference between MF-ASC and GP_LAPL per researcher in that case, color-coding the measured (absolute) correlation between high- and low-fidelity. MF-ASC clearly outperforms GP_LAPL when the absolute correlation is greater than 0.75, consistently with our observations on simulated user interests. This result is due to the sensitivity of MF-ASC to noise in low-fidelity data. In a passive learning setting, the multi-fidelity models perform not worse than the single-fidelity ones [11, 28], as the former are reducible to the latter. However, in active learning, low-fidelity noise affects the search on high-fidelity data.



(A) Regret vs budget (Λ). Solid curves show the average and dashed curves show the 0.2- and 0.8-quantiles.



(B) Difference between regret of MF-ASC and GP_Lapl.

FIGURE 3.5: ACL with real-user intended sets.

3.5.4 Assessing selection strategies

We now proceed to an exhaustive experimental evaluation with simulated user data. We first assess our choice of acquisition criterion vs. the σ-optimality criterion of GP-SOPT [60]. We set GP-SOPT against GP_Lapl [78], which uses the UCB criterion that we have adopted in MF-ASC. We set the value of β_t, which controls the exploration-exploitation tradeoff in GP_Lapl, to 0.01, as we reported in section 3.5.2. Figure 3.6 shows the results on MAG-Sm, tuning the GP-SOPT α and k parameters. Notably, GP_Lapl, represented by a black curve, outperforms all GP-SOPT variants, represented by solid-color curves. This result establishes that σ-optimality is inappropriate for regression problems. Henceforth, we use GP_Lapl as the state-of-the-art benchmark combining the best practices of the previous works.

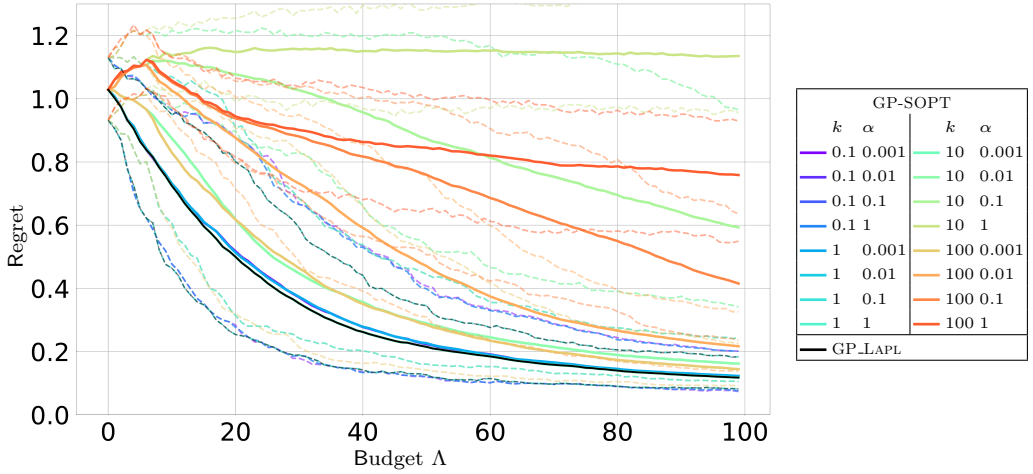


FIGURE 3.6: Relative regret of GP-SOPT compared to GP_Lapl on MAG-Sm for different k, α. Solid curves show the average and dashed curves show the 0.2- and 0.8-quantiles.

3.5.5 Effects of varying the fidelity ratio

Next, we study the effect of the budget Λ and fidelity ratio r on the regret of MF-ASC. Figure 3.7 presents our results on MAG-Sm. Expectedly, the budget Λ, i.e., the number of high-fidelity user evaluations, affects quality, achieving a 5-fold improvement from 10 to 100 queries. The ratio of low-fidelity queries per high-fidelity evaluation also has a significant though gradually attenuated effect. This result justifies our choice of $r = 5$ as a default value that achieves a reasonable quality-speed tradeoff.

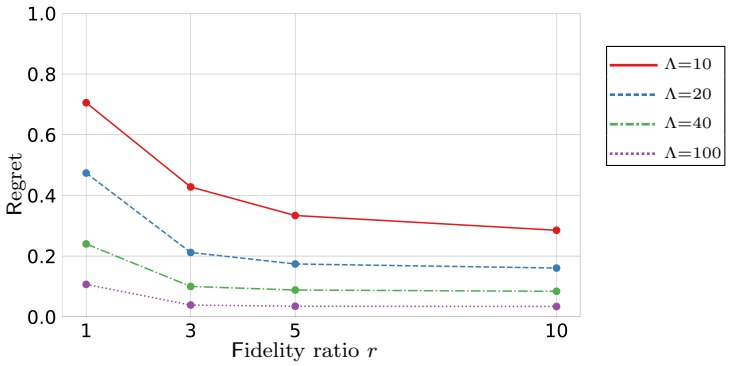


FIGURE 3.7: Relative regret on MAG-Sm vs. r.

3.5.6 Effect of budget

We now commence our comparative evaluation of different methods, studying the effect of the interaction budget Λ on performance. Figure 3.8 shows our results on average relative regret and Recall@Λ (the share of found elements in the intended set) vs. Λ on all datasets, setting MF-ASC against the best method derived from the state of the art, GP_LAPL, and the LABELPROP baseline. We consider an ideal scenario when low-fidelity is equal to high-fidelity. MF-ASC consistently outperforms both competitors with a significant improvement over GP_LAPL on all datasets. On the other hand, LABELPROP, being a passive method, shows the worst performance. We now commence our comparative evaluation of different methods, studying the effect of the interaction budget Λ on performance. Figure 3.8 shows our results on average relative regret and Recall@Λ (the share of found elements in the intended set) vs. Λ on all datasets, setting MF-ASC against the best method derived from the state of the art, GP_LAPL, and the LABELPROP baseline. We consider an ideal scenario when low-fidelity is equal to high-fidelity. MF-ASC consistently outperforms both competitors with a significant improvement over GP_LAPL on all datasets. On the other hand, LABELPROP, being a passive method, shows the worst performance.

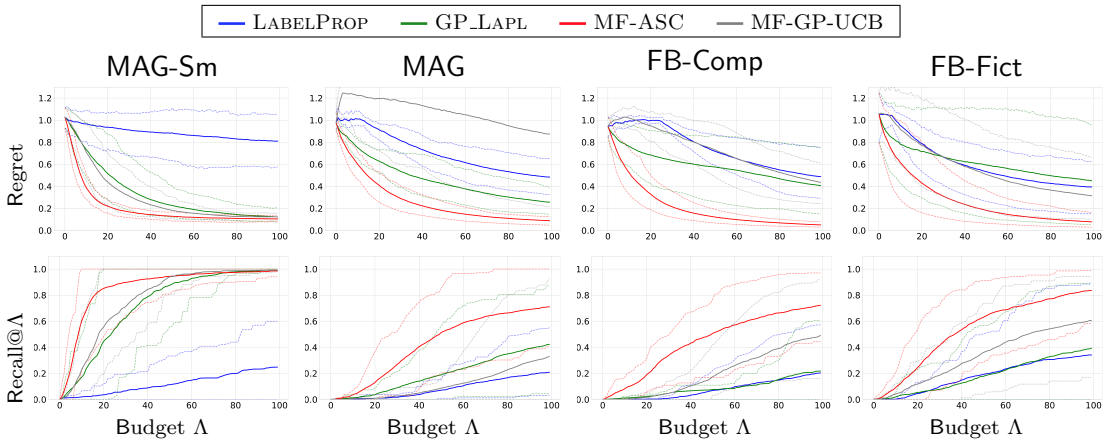
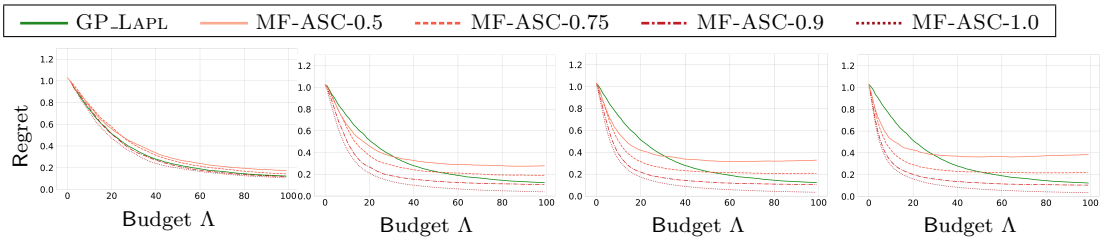


FIGURE 3.8: Relative regret and Recall@Λ vs. Λ in case low-fidelity is equal to high-fidelity. Solid curves show the average and dashed curves show the 0.2- and 0.8-quantiles.

3.5.7 Sensitivity to fidelities correlation

Thus far, we have used a low-fidelity function correlated to the simulated user input. Here, we study the sensitivity to the correlation between the low- and high-fidelity functions. Figure 3.9 reports our results vs. different handcrafted correlation values and also different fidelity ratios r, ranging from r=1 (Figure 3.9a) corresponding to one low-fidelity evaluation after each high-fidelity request, to r=10 (Figure 3.9d). The results indicate that with a reasonably high correlation value MF-ASC outperforms GP_Lapl, while with lower values it does not fare so well. Besides, the advantage of MF-ASC becomes evident at earlier interaction steps as more low-fidelity evaluations are performed (Figure 3.9c).

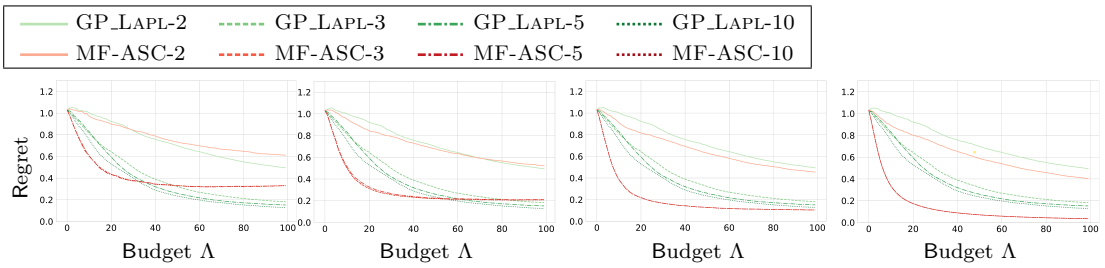


(A) Fidelity ratio $r = 1$ (B) Fidelity ratio $r = 3$ (C) Fidelity ratio $r = 5$ (D) Fidelity ratio $r = 10$

FIGURE 3.9: Relative regret vs. Λ and fidelity correlation (shown by numbers in the legend), MAG-Sm.

3.5.8 Sensitivity to score granularity

Our simulated user relevance score has hitherto been continuous. Still, real-world users provide a granular input of a few grades [85]. We now examine the sensitivity of our results to the granularity of the user input. Figure 3.10 reports our results with the



(A) correlation 0.5 (B) correlation 0.75 (C) correlation 0.9 (D) correlation 1.0

FIGURE 3.10: Relative regret vs. Λ and number of discretized score values (shown by numbers in legend) on MAG-Sm.

simulated user input discretized, with both MF-ASC and GP_Lapl, for different granularities of this discretization as well as for different correlation values between high- and low-fidelity scores. Note that regret is still calculated with respect to continuous high-fidelity values. We also use continuous values for the low-fidelity scores as they are specified by an internal model of the user which is expected to work with such continuous values.

Notably, a discretization granularity of 2 levels (i.e. binary scores) gives poor quality, whereas 3 levels achieve quality indistinguishable from that of 10 levels; that is also virtually identical as that of the continuous case, which we omit from the figure as its difference from the 10-level one is imperceptible. These results confirm that a continuous user function, which we have used in the rest of our simulated-user studies, yields results that we can also achieve with the kind of discrete input that real-world users provide.

In the complementary problem of function interpolation, the benefits of using a multi-fidelity approach over a single-fidelity one were studied in [17]. That study shows that, depending on the cost ratio of the high-fidelity function to the low-fidelity function and the correlation between them, the multi-fidelity approach is beneficial, whereas when the cost ratio or correlation is sufficiently low, multi-fidelity loses its advantage.

3.5.9 Scalability

Last, we evaluate MF-ASC on its ability to produce real-time answers with growing budget Λ. Figure 3.11 shows our results on time per user interaction (i.e., high-fidelity evaluation) vs. Λ, as well as its variance, on logarithmic axes. As expected (cf. Section 3.3.3), the time per interaction of MF-ASC grows cubically in the number of high-fidelity evaluations. As MF-ASC evaluates r=5 times more nodes than GP_Lapl, due to the low-fidelity internal evaluations, it incurs additional computational overhead. However, this time remains within real-time performance. Even with the biggest dataset, FB-Comp, time per interaction is less than 1 second.

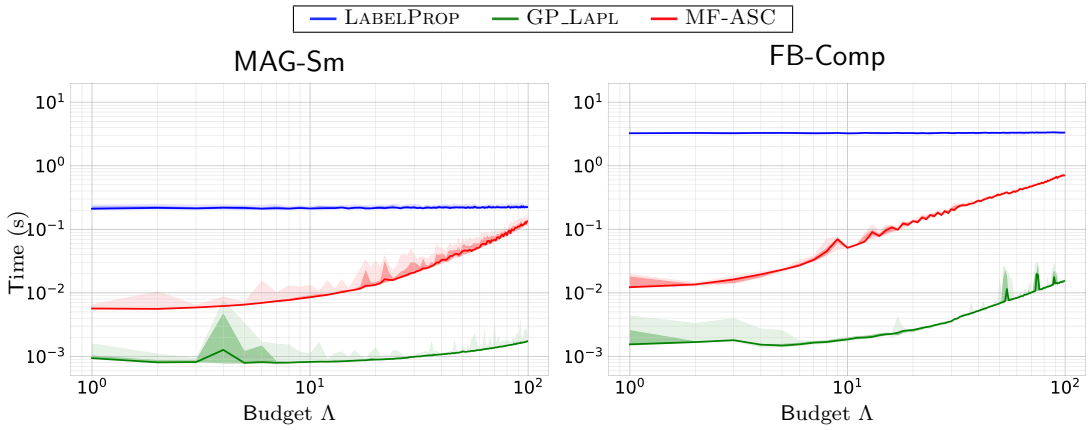


FIGURE 3.11: Time per iteration; shades denote 0.8 quantiles.

3.6 Conclusions

We proposed MF-ASC, an active search mechanism that integrates the user feedback with information from an *internal evaluation function* via *co-kriging* Gaussian interpolation, and thereby finds the results the user wants with the minimal user input. As MF-ASC performs similarity computations in a preprocessing step, it faces no scalability obstacle — its time complexity is cubic only in the size of the set of *scored* nodes; the core motivation of this work is to ensure that this set is kept small. Our experimental study on the real and simulated user data shows that MF-ASC surpasses the state of the art, delivering highly relevant information after a few user interactions.

Chapter 4

Industrial applications of multi-fidelity classification and active search

This chapter describes three real projects and demonstrates the application of Bayesian optimization and multi-fidelity active search and classification methods within them. Section 4.1 shows the application of Bayesian optimization of a low-fidelity function for the engineering design of the muon shield. Section 4.2 (in particular, 4.2.2.5 and 4.2.3.3) demonstrates the application of multi-fidelity classification from chapter 2 to improving quality of datasets for the data-driven rock-type identification project. Finally, section 4.3 (in particular, 4.3.4 and 4.3.5.5) shows how the multi-fidelity active search algorithm from chapter 3 can be employed for reducing the costs of datasets annotation in the directional drilling accidents prediction project.

4.1 Active muon shield optimization

The SHiP[1] (Search for Hidden Particles) experiment [86] is designed to search for very weakly interacting particles beyond the Standard Model [87] which are produced in a 400 GeV/c proton beam dump at the CERN SPS. The critical challenge for this experiment is to keep the Standard Model background level negligible. In the beam dump, around 10^{11} muons are produced per second. The muon rate in the spectrometer has to be reduced by at least four orders of magnitude to avoid muoninduced backgrounds.

[1] `https://ship.web.cern.ch/`

The muon shield is a critical component of the SHiP experiment, which deflects the high flux of muons produced in the target, that would represent a very serious background for the particle searches, away from the detector. The shield consists of eight magnets and each magnet is parameterised by seven geometric values: length, width, etc.

Since the cost of the muon shield is significant, our aim was to find the most efficient solution at a lowest cost possible. The main goal of our research [88] was to find a light and efficient magnetic shield. In order to achieve this, we applied a Bayesian optimisation algorithm based on Expected Improvement criterion.

4.1.1 Data

For the evaluation of the shield performance we use 18 million simulated events passed through the muon shield and measured at a scoring plane 64m downstream of the muon shield at the z of the first tracking station of the spectrometer. The transverse (x, y) position of muons is obtained at the first tracking station.

4.1.2 Optimization problem

The loss function depends on the physical performance of the shield and its weight as a proxy for the cost.

$$L(W, \Sigma) = (1 + \Sigma) \left(1 + exp\left(\frac{W - W_0}{W}\right)\right),$$

where $\Sigma = \sum_\mu \sigma_\mu$, σ_μ is an importance weight of the muon [4, 88], W is a weight of configuration and W_0 is a weight of a baseline configuration [4] shown in Fig.4.1a. The weight of the baseline is about 1900 tons and Σ for the baseline is approximately equal to 32.

The following problems are addressed during the optimisation: (i) Optimisation is performed in 42-dimensional space (ii) Computation of the Σ is time consuming. (iii) Computation of the Σ is noisy due to limited statistics in Monte - Carlo simulations.

4.1.3 Solution

To ensure a more even coverage of the muon phase space, the number of muons is capped in bins of momentum and transverse momentum, and augmented through resampling available muons in bins with low occupancy. This also reduces the overall number of

muons to be simulated to about 500 thousand, and helps us to decrease the time of computations by factor of 8 times in average.

In order to optimize the loss function, we applied a Bayesian optimization approach based on surrogate modeling with Gaussian processes.

Several conditions on the family of priors that ensure convergence of the Bayesian optimization to the optimum are specified in [89], they are satisfied by a Gaussian process with a constant mean function. Therefore, the convergence is guaranteed when the principle of the minimum expected risk is used, as, for instance, the expected deviation from the global optimum:

$$x_{i+1} = \underset{x}{\operatorname{argmin}}\ \mathbb{E}[\|f(x) - f(x_*)\||D_i] \tag{4.1}$$

Expression 4.1 is computationally expensive since it requires estimation of $f(x_*)$, thus in the present work we replace this criterion with its common approximation called *Expected Improvement* [90]:

$$x_{i+1} = \underset{x}{\operatorname{argmin}}\ \mathbb{E}[\max\{0, f(x) - f(x_+)\}|D_i], \tag{4.2}$$

where $x_+ = \underset{j=1..i}{\max}\ y_i$ Expected Improvement has also been shown to converge under additional mild assumptions [91], but its main advantage is a closed-form expression, that doesn't require numerical integration:

$$\mathbb{E}[\max\{0, f(x) - f(x_+)\}|D_i] = \hat{\sigma}_N^2(x) Z \Phi(Z) + \hat{\sigma}_N^2(x)\phi(Z), \tag{4.3}$$

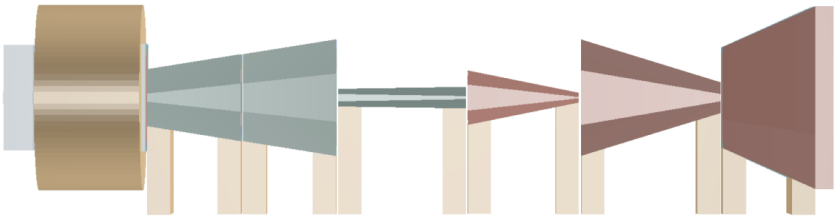
where $Z = \frac{\hat{f}_N(x) - f(x_+)}{\hat{\sigma}_N^2(x)}$, ϕ and Φ are PDF and CDF of the standard normal distribution respectively.

4.1.4 Results

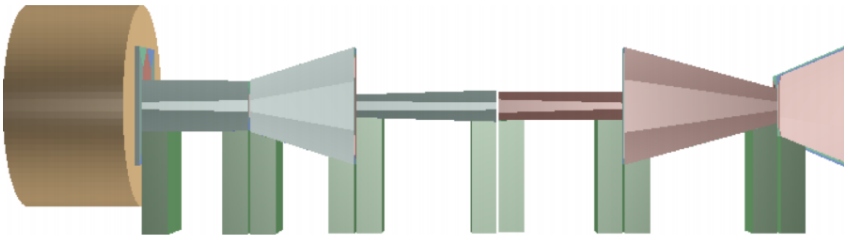
The optimisation procedure is stopped after 5000 iterations. The obtained configuration is found to be lighter by 25% than the baseline while having about the same rejection capability as illustrated in Fig. 4.2. Fig. 4.1b illustrates the new found configuration.

4.1.5 Conclusion

Bayesian optimisation is a powerful tool which can be applied for the optimisation of non differentiable functions. Usage of the additional low-fidelity source of information can speed up the whole process of optimization, meanwhile high-fidelity source (expert decision or extensive simulation) is crucial for the final validation. We have demonstrated



(A) The baseline.



(B) The best found.

FIGURE 4.1: The SHiP muon shield shield configurations, see the description in [4].

that this method can be successfully applied even to the complicated real optimisation problems in physics.

4.2 Rock type identification for directional drilling.

Oil and Gas reserves are becoming more complex for an efficient exploration with significant financial margins nowadays. There is a number of examples when oil companies have to approach thin oil/gas bearing layers of complex topology. These layers, or the target intervals, can be as thin as a couple of meters. One of the common ways of exploring such intervals is directional drilling [92].

The directional drilling aims to place a wellbore in a way that it has the maximal contact with the thin target layer. The latter requires the wellbore trajectory to follow all the folds of the layer as accurate as possible. To follow the folds, drilling engineers use Logging While Drilling (LWD) data recorded by physical sensors placed on a borehole assembly 15 m to 40 m behind the drilling bit. The sensor data is the source of information on whether the sensors are within the oil/gas bearing formation or not. Based on this information, engineers correct the drilling trajectory.

The gap between the bit and the sensors is a significant issue preventing the timely correction of the drilling trajectory. It can result in a non-optimal placement of the well

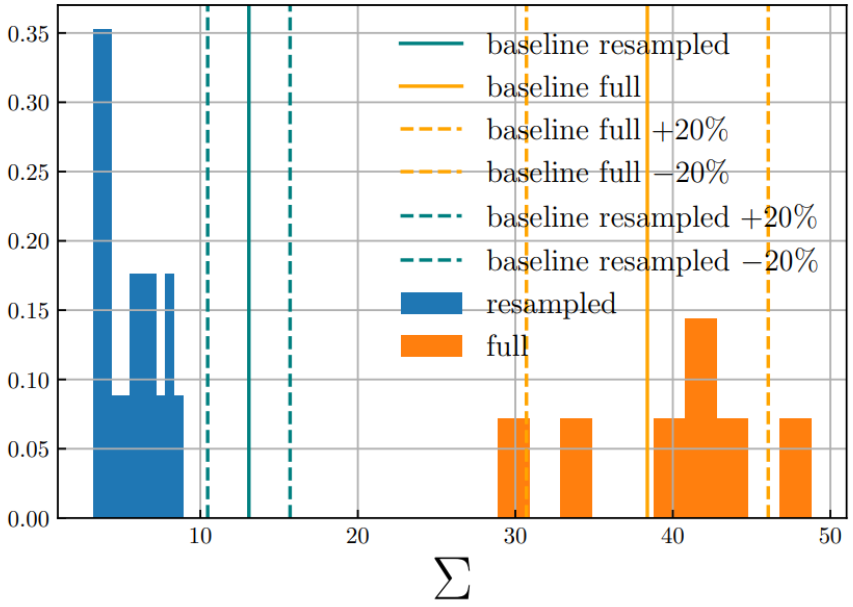


FIGURE 4.2: Distributions of rejection performance scores for resampled (low-fidelity) and full (high-fidelity) muon samples for the best found shield configuration, comparing with the baseline solution.

or multiple cost-intensive re-drilling exercises. Figure 4.3 shows schematic illustrations to supply the definition of the problem.

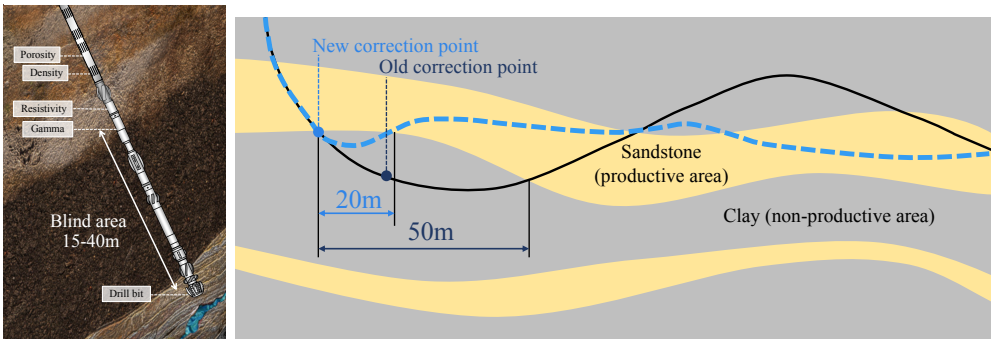


FIGURE 4.3: Schematic illustration of the drilling string (on the left) and the effect of timely applied trajectory correction (on the right): the black curve shows a trajectory in case rock types are available only at the distance of 15 m from the drilling bit, blue dashed curve corresponds to the trajectory when rock types are available at the drilling bit.

This work proves the feasibility of a technology aimed at optimizing trajectories of directional wells ensuring best possible contact of the wellbore and the target layer of the reservoir. The technology allows tackling the challenge of a delayed reaction on trajectory correction during drilling the directional wells. Machine learning allows eliminating 15 m to 40 m gap between the drilling bit and LWD sensors, and corresponding speeding

up the decision making process at the trajectory correction. Along with machine learning approaches we examine how mathematical modeling can advance machine-learning based approaches.

Basically, a trained data-driven algorithm allows a computer to identify the moment when the bit touches a shale-rich part of the formation by a continuous screening through the real-time Measurements While Drilling (MWD) data. In machine learning, this problem is referred to as the two-class classification problem: we need to create a predictive classification model (a classifier) that can identify whether the bit at the current moment is in the shale-rich part of the formation (the first class) or not (the second class). In addition to labeling objects, the classifier can output the probability of the object to belong to a specific class, thus, allowing to introduce confidence of the predictions.

From the machine learning perspective, the main peculiarities of the problem are missing values in measurements and a relatively high imbalance of classes: there are only 13.5% of shales and hard-rocks in the available data, where "hard" refers to a measure of the resistance to the localized plastic deformation induced by either mechanical indentation or abrasion, and 86.5% of sands. Therefore, we tested different machine learning algorithms under these peculiarities, and developed appropriate evaluation methods of their performance.

The main contribution of this part of work is a novel data-driven approach for identifying a lithotype at the drilling bit. We prove the feasibility of this approach by studying mathematical and physical modeling and applying three essential machine learning baselines (Logistic Regression, Neural Networks and Gradient Boosting on Decision Trees) for the problem of lithotype classification based on MWD data, which come from 27 wells of the Novoportovskoe oil and gas condensate field on Western Siberia.

4.2.1 Machine Learning in drilling application

There are previous studies on the involvement of machine learning for detection of a material type at drilling bit. The work [93] covers an analysis of the applicability of the regression and classification based on Gaussian Processes and unsupervised clustering for on-bit rock typing with MWD data. In the report the authors consider the rate of penetration (ROP), a weight on the bit (WOB), and top drive torque (TRQ) as the key parameters for building the data-driven forecasting model. One of the conclusions is that a value called adjusted penetration rate (APR) (Eq. 4.4) is the best reflection of a features specifics of the rock which is unknown a-priori. The authors conclude that the optimal way to predict a rock type at the drilling bit is to apply a hybrid model combining the advances of both supervised classification and unsupervised clustering.

$$\mathrm{APR} \propto \frac{\mathrm{ROP}}{\mathrm{WOB}\sqrt{\mathrm{TRQ}}} \tag{4.4}$$

APR is tested in this study as well as another characteristic utilized by many authors [93, 94], the Specific Energy of Drilling (SED):

$$\mathrm{SED} = \frac{\mathrm{WOB}}{A} + \frac{2\pi \times \mathrm{RPM} \times \mathrm{TRQ}}{A \times \mathrm{ROP}}, \tag{4.5}$$

where A represents a cross section area of the wellbore.

[94] illustrates that unsupervised learning together with the minimization of SED is a promising approach for the optimization of the penetration rate. Another effort on penetration rate optimization is presented by [95]. The authors use the Random Forest algorithm to build a model linking the penetration rate with weight on the bit, rotation speed, drilling mud rate, and unconfined rock strength. The model allowed to optimize the penetration rate for up to 12% for the wells close to the ones in the training set.

[96] and [97] describe an application of Artificial Neural Networks for material typing and rock typing at drilling. MWD-like measurement and the trained Neural Networks allow a relative classification error to be as small as 4.5% for the case with the complete set of available mechanical measurements (features).

Gaussian processes and neural networks are not the best fit for the rock type classification with MWD data in production as they can not automatically handle missing values that typically occur in MWD data. Thus, both methods require training data without missing values that implies the development of accurate imputation procedures, however, such procedures can typically be reliable only during post-processing of data. The difference between these two types of models is in the preferred data size and its dimensionality. Gaussian processes are based on the Bayesian approach, so they can work fast only when the training sample is small, moreover, this method struggles with high input dimensions. In contrast, neural networks can work well in large dimensions and fast for large samples, but they are not suitable for small datasets. In case we need to reflect the temporal behavior of MWD in input features, we face high dimensions, also for real-life MWD sample sizes are large. Therefore, for the production-grade model, neural networks would be more preferable than Gaussian processes, if there are no missing values in the training and real-life data.

Decision trees and methods based on them [98] such as Random Forest and Gradient boosting can automatically handle missing values, and they are comfortable with large sample sizes. However, the tree-based methods are weak at data interpolation, so they generalize well only when the density and the diversity of points in the training sample are high. Gradient boosting can also handle classes imbalance by automatic weighting the importance of data entries while maximizing the quality of a classifier.

4.2.2 Data description and pre-processing

This section specifies the origin of data used in our work and its pre-processing procedures.

4.2.2.1 Geological formation of the interest

The Novoportovskoye oil and gas condensate field, located within the Yamal Peninsula, 30 km from the Gulf of Ob Bay, is the largest field under the development of the northwest of Siberia, Russia. The formation includes several strata, the most productive of which is the Lower Cretaceous NP-2-3 — NP-8 (the formation depth is 1800 m), and sand layers of the Tyumen suite J-2-6 (the formation depth is 2000 m). The reservoir rocks are fine-medium grained sandstones and siltstone with thin layers of shales and limestone. The average rocks permeability is 0.01-0.03 μm^2 and the porosity is about 18%.

4.2.2.2 Initial data

The initial data included mud logging, involved the rig-site monitoring and assessment of information measured on the surface while drilling and MWD, LWD data from downhole sensors. The main purpose of MWD systems is to determine and transmit the inclinometry data (zenith angle and magnetic azimuth) to the surface in real time while drilling. It is necessary to determine the well trajectory. Sometimes the inclinometry data are supplemented with information about the drilling process and logging data (LWD). Logging allows to measure the properties of the rock, dividing the geological section into different lithotypes.

The data includes the following parameters: WOB, TRQ, ROP, APR, SED, also rotary speed (RPM), input flow rate (Q in), output flow rate (Q out), standpipe pressure (SPP), and hook load (HL).

The initial information about the drilled lithotypes was Lithology Map produced by petrophysical interpretation of LWD measurements which were represented by natural gamma radiation; apparent resistivity; polarization resistance; electromagnetic well log; induced gamma-ray log; neutron log; acoustic log.

LWD petrophysical interpretation was also used to compare the real values of the lithotype and the prediction obtained.

4.2.2.3 Pre-processing

The pipeline for the data preprocessing consisted of four main steps: extraction of required columns from the raw data files; selection of the relevant horizontal parts of the wells; merging data from different sources; unifying depths steps for the constructed dataframes.

After preprocessing the raw data, we reduced the granularity of time-series by aggregating them over depth intervals of 0.1 meters. For intervals containing any data, we averaged its values, for intervals without data, we used forward fill with a constant that equals the latest preceding value.

4.2.2.4 Feature engineering and selection

In this section we describe several methods of refining information about rock types which is stored in MWD and LWD data, so that classifiers can take advantage of it.

At each moment of time not only current MWD and LWD values characterize the type of rocks, but also previous values and their relationships among each other bring additional information. Therefore, in this section, we start with considering a few ways to incorporate such information as input features.

The *Basic* (B) set of features used in a predictive model includes original mechanical features, SED, and APR. We also derived new features from the basic ones:

- *Derivatives* (D) are the rolling mean and standard deviation with the window size of 1 m, and the difference between values on rolling window's borders;
- *Lagged* (L) are the basic features with their values of 0.1, 0.5, 1 and 10 m back;
- *Fluctuations* (F) are the standard deviation of original time series inside the aggregated (see sec. 4.2.2.3) intervals of 0.1 m;

- *Extra* (E) features are the true class values 20 and 50 m back, since they can be obtained from LWD measurements with such spatial lags.

Generating too many interrelated features results in their redundancy, longer time of models training and risk of overfitting. Thus, after feature engineering, we ran the feature selection procedure which had the aim to select the subset of features that maximized classification quality.

We used a "greedy" approach for feature selection: the procedure started from the empty set and expanded it by adding, step by step, the most impactful feature from the pool of remaining ones according to the selected quality metric.

4.2.2.5 Rock type labels refining with multi-fidelity models.

Labels refining follow up the discussion in the original paper [99] about improving classification quality with multi-fidelity methods.

The rock types annotation is conducted by geophysicist based on LWD data. The main issue with such labels is presence of bias in human expertise, because there are no clear criteria to attribute cases with mixed rock types to a particular class. For instance, we have conducted several tests with re-annotation of the same dataset by different experts and found the difference in labels over around 1 percent of the total wells length. Moreover, in some cases, the difference between experts' annotations on particular wells were up to 20 percent of the wells length.

In this exercise, we are using the following approach to labels refining based on the multi-fidelity classification. Let's consider the experts' annotations as a low-fidelity data source. Then we arrange a classifier to predict these labels on a validation set using LWD data and treat such predictions as a high-fidelity source, which can still deviate from the ground truth, but with much less amount of errors. The rationale behind this idea is the following: assuming that a classifier can generalize well to new data and having a dataset annotated by multiple experts, that give unbiased annotations on average with respect to the ground truth, we are likely to obtain less biased predictions on the validation set. Finally, we use a multi-fidelity classification method (see 2) to fuse expert annotations with high-fidelity estimates of the labels.

4.2.3 Results

In this section we:

- report on how different sets of features affect the quality of the rock type classification, which features are more informative;
- examine selection of hyperparameters for different machine learning methods;
- compare the performance of different machine learning methods and show how classification quality depends on the balance of classes;
- report on labels refining procedure results.

4.2.3.1 Feature selection results

For feature selection we used ROC AUC quality metric obtained via leave-one-well-out cross-validation (LOWO-CV). Since sensors readings are autocorrelated, it is crucial to split the dataset by wells, not by random subsets during cross-validation. Otherwise, data leakage will take place resulting in overestimated quality, that is, models will have more information about the test set during cross-validation than they will have in the field test on the new wells.

The classifier was constructed with gradient boosting of 50 decision trees, each of maximal depth 6. The best selected set *Greedy* (G) consists of ROP, HL, rolling differences of WOB, 1m rolling standard deviations of ROP and TRQ, 1m moving average of ROP, 0.5 meters lagged TRQ, and 10 meters lagged Q out, Q in, HL and TRQ.

We also fine-tuned gradient boosting hyperparameters by increasing the number of decision trees up to 100 and conducting a grid-search LOWO-CV for maximal depth of the trees, random subspace share and sub-sampling rate. Table 4.1 summarizes the results of the feature selection process. We obtained the best results for all quality metrics using the selected set of features G along with extra set E. In particular, Accuracy L is larger than 0.9.

Feature set	ROC AUC	PR AUC	Accuracy L
-	0.494	0.181	0.866
B	0.794	0.492	0.865
B, F	0.803	0.484	0.867
B, F, D, L	0.829	0.504	0.870
G	0.850	0.559	0.888
E	0.653	0.359	0.879
B, E	0.848	0.581	0.900
B, F, D, L, E	0.870	0.600	0.902
G, E	**0.878**	**0.614**	**0.905**
G, E (fine-tuned)	**0.880**	**0.625**	**0.910**

TABLE 4.1: Feature selection results. Greedy selected set of features combined with the Extra set provides the best quality.

Figure 4.4 shows the dependence of quality metrics on learning rate and the number of trees in the ensemble obtained by gradient boosting. Low learning rates (blue curves) result in underfitting, whereas high learning rates (red curves) result in overfitting of the model. Orange and green curves correspond to a good trade-off.

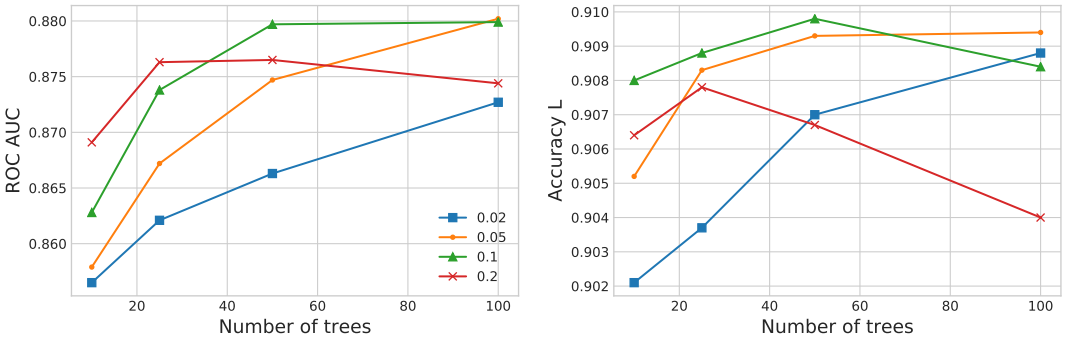


FIGURE 4.4: Quality vs Gradient Boosting parameters. Curves of different colors correspond to different learning rates.

Figure 4.5 shows feature importances for the fine-tuned classifier trained on the whole dataset. Importance scores indicate how many times a particular feature played the key role in the classifier's decision.

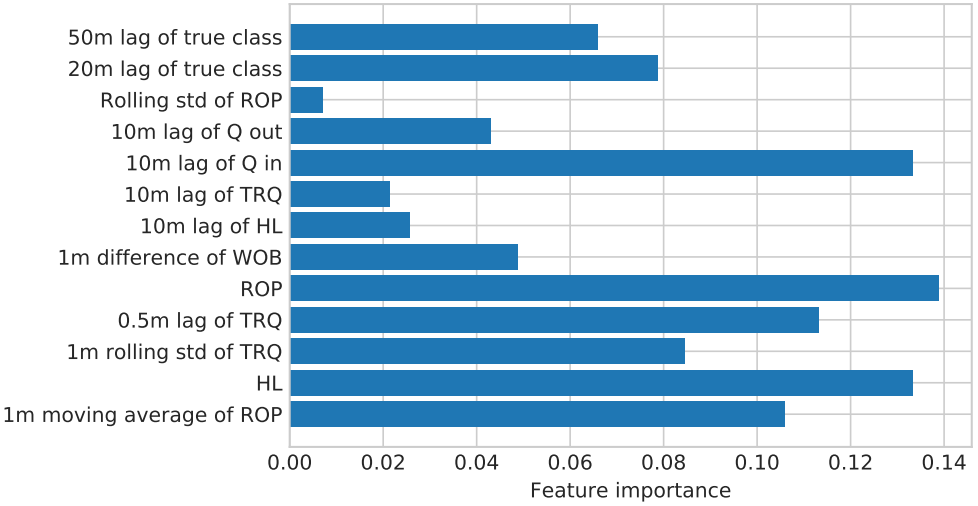


FIGURE 4.5: Importance of features for the gradient boosting classifier predictions. Two sets of features are included: Greedy and Extra. The bottom-up order of Greedy features corresponds to their selection order during the selection procedure.

4.2.3.2 Algorithms performance

We compared three classes of machine learning methods in detail: Logistic regression, gradient boosting, and neural networks. The results in this section correspond to the performance of the best-found configurations for each method using LOWO-CV. All methods used both Greedy and Extra sets of the features.

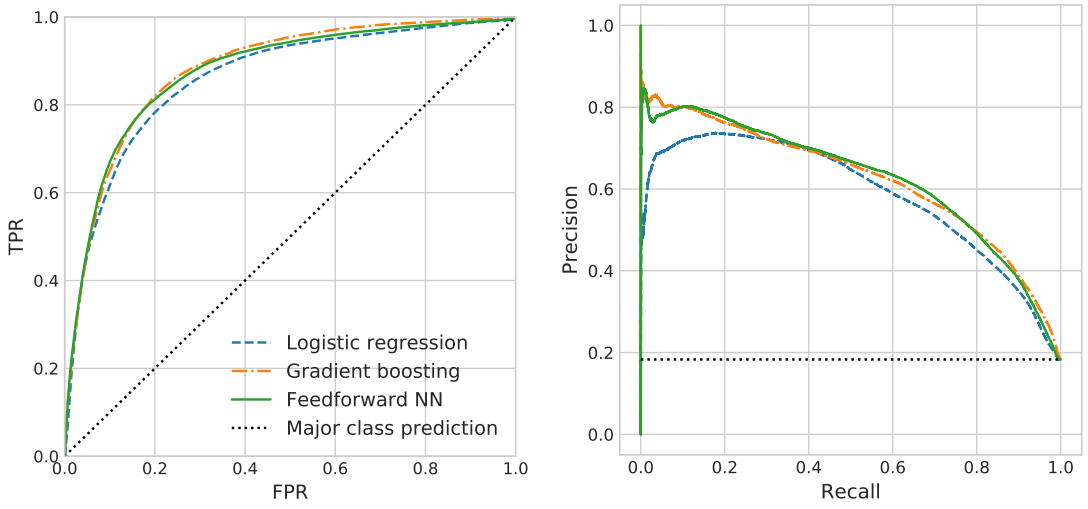
For logistic regression, we observed the best quality when no regularization is applied. The best-found configuration of gradient boosting for 100 trees had the following hyper-parameters: learning rate 0.05, maximal depth 3, random subspace share 0.8, and sub-sampling rate 0.55. For Feedforward neural networks (NN) we tested different architectures with 2-, 3- and 4-layer networks. The best found configuration had two hidden layers with 100 and 500 neurons using ReLU activation between layers.

Table 4.2 summarizes the best performance of different classification methods. Gradient boosting uniformly dominates logistic regression, in turn, feedforward NN and gradient boosting qualities are comparable due to the preprocessing pipeline we developed, which filled the missing data sections with rather adequate values. LSTM training time was impractically long, whereas its best-found performance was similar to feedforward NN.

Algorithm	ROC AUC	PR AUC	Accuracy L
Always predict the major class	0.494	0.181	0.866
Logistic regression	0.860	0.585	0.908
Gradient boosting	**0.880**	**0.625**	0.910
Feedforward NN	0.875	**0.625**	**0.911**

TABLE 4.2: Performance of machine learning approaches logistic regression, gradient boosting, and feedforward NN. All performance measures are better if higher.

Figures 4.6 present visual comparison of performance of different classification methods.



(A) ROC curve (B) PR curve

FIGURE 4.6: Performance curves for three different machine learning approaches: logistic regression, gradient boosting, and feedforward NN; compared with the input-agnostic method that always predicts the major class. As the curves for gradient boosting and feedforward NN lie higher than the curves for logistic regression, we conclude that the corresponding models are better.

Figure 4.7 shows performance of the gradient boosting with respect to lithotype classes balance. The lithotype predictions with the trained classifier are better than major-class predictions for 24 out of 27 wells. Improvement of Accuracy L increases if the classes are more balanced, that is, if they tend to have more equal shares of shales and rocks (first class), and sands (second class). However, the improvement varies significantly from well to well. Figure 4.8 shows examples of lithotype classification on three wells with different achieved quality.

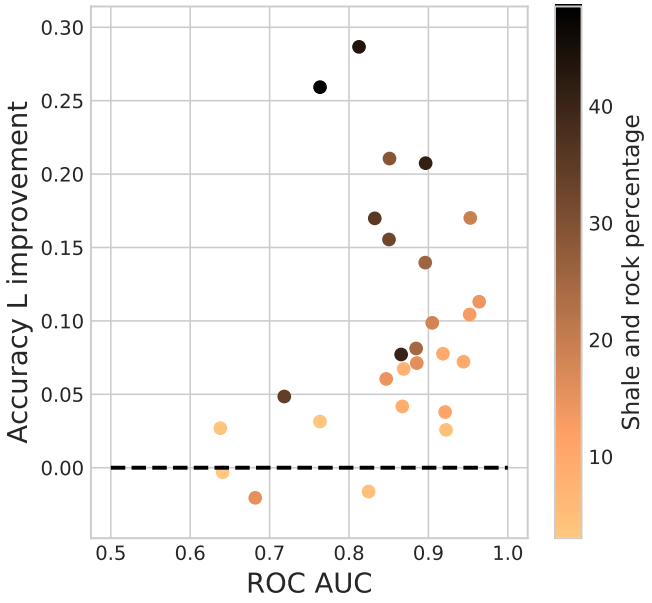
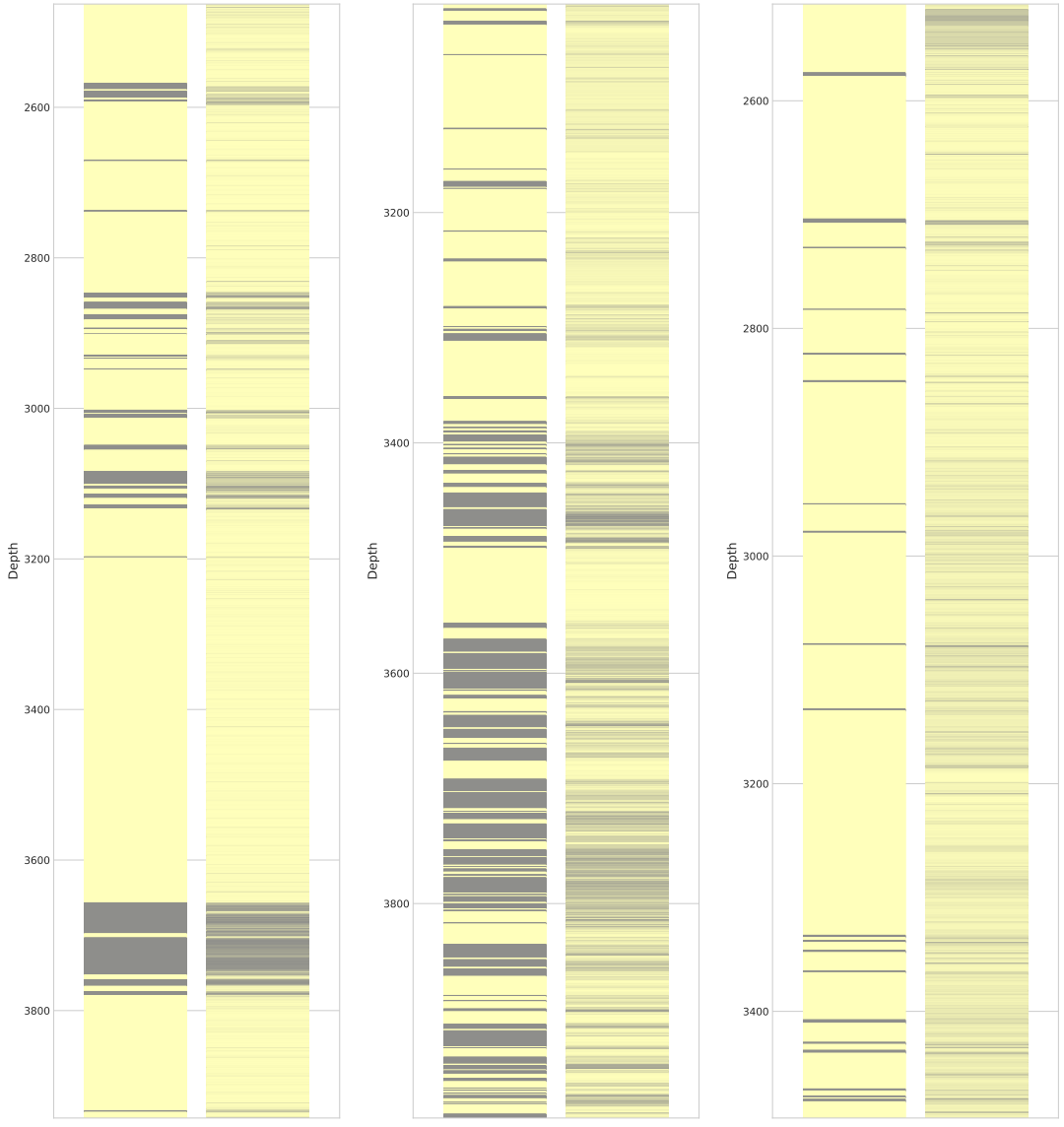


FIGURE 4.7: Gradient boosting performance on different wells with respect to well-specific shale and rock percentage. The vertical axis represents the improvement of Accuracy L from using gradient boosting over the major class predictions.

4.2.3.3 Labels refining results

In these experiments we used three features of LWD data: gamma ray logs, density logs and average value of imager channels. Since we don't have the real ground truth data, we will refer to already refined labels as (pseudo) ground truth. To demonstrate the stability of the reconstruction method, we distort the ground truth labels by assigning fake rock types to random intervals on wells data; the amount of distortion is controlled by the length of fake intervals. Figure 4.9 shows the amount of errors (percentage of the total wells length) remaining after the labels refining procedure with respect to the initial noise, it also shows the amount of errors in high-fidelity labels obtained at the intermediate step. Although high-fidelity labels already have much less noise than the distorted ones, this experiment shows that multi-fidelity classification can further reduce the amount of errors by a significant margin.



(A) ROC AUC 0.9657 (B) ROC AUC 0.8004 (C) ROC AUC 0.6298

FIGURE 4.8: Examples of lithotype classification for three wells with different achieved quality: from one of the best on the left through average in the middle to one of the worst on the right. In each subfigure the left column shows the true lithotype values: yellow color represents sand, grey color represents shales and hard-rock; the right column shows the respective probability of lithotypes given by the classifier.

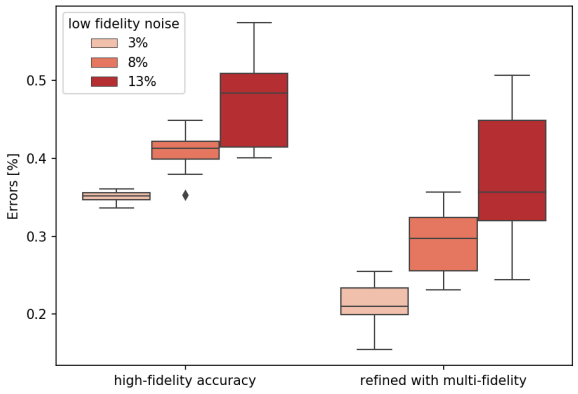


FIGURE 4.9: Errors of labels refining methods at different distortion levels (noise). Box plots show variety of results across 10 random seeds.

4.2.4 Conclusion

This study illustrates the capabilities of machine learning to handle the real technological issues of directional drilling. The accuracy of prediction of rock types relevant to directional drilling management reaches 91%, that is, the classification error drops from 13.5% (major-class prediction) down to 9% (the best-achieved performance by the examined algorithms). The involved algorithms allow real-time implementation which makes them useful for drilling support IT infrastructure.

In the thesis, we have additionally elaborated on the problem of sample labels correction discussed in the original paper. The methodology of how to apply multi-fidelity classification for labels correction was proposed and experimentally demonstrated to drastically reduce noise in labels caused by expert-specific bias.

4.3 Search of similar accidents cases for directional drilling.

Anomaly detection is the identification of rare objects, events, or observations that are significantly different from most data [100]. Regardless of the level of the well's construction technology, anomaly situations inevitably happened during drilling. Anomalies may have both positive and negative influence on a system, depending on their interpretation and consequences. For example, a significant increase in the number of visits to a website might be considered as a positive anomaly, that results in website popularity growth. In case of directional drilling, abnormal behavior rather leads to failures (emergencies which make any further work impossible or delay future activity), than to the improvement of the drilling process.

Accidents have a significant impact on the further operation of wells and usually lead to an increase in construction time and the cost of work. The drilling support engineers use mud logging to detect accidents while drilling. Due to the fact, that engineers support a large number of wells at the same time, they usually do not have time to monitor all the wells online, and drilling accident patterns are considered after the occurrence of the accident. Thus, the creation of a system that signals about drilling accident will help engineers more efficiently support the drilling process. Early detection of failures can significantly reduce the nonproductive time of the well associated with the elimination of the accidents consequences and costs for additional materials and technical resources. Most of the oil and gas companies also create knowledge base systems, in which information about the failures is collected and carefully studied in order to use accumulated experience for further failures detection by comparison the current drilling conditions and previous cases. Such an approach is called analogues search and was successfully used for time-series forecast [101, 102]. Thus, the presence of extra support during drilling operations as an expert system, that includes accumulated accident detection experience, is an effective method for making the right real-time decisions. Such a system will allow to avoid additional expenses during drilling operations, and reduce the high workload level on the drilling engineers. Therefore, the development of the methods, that can help to detect failures during the real-time drilling operations is essential for the oil and gas industry.

This work is aimed at the analysis of the analogues search approach for application to drilling accidents detection. In more detail, the key objectives of the article are the development of the model, that will be able to distinguish similar and non-similar drilling situations, the analysis of model's applicability limits, and quality.

The main contribution of this part of work is an anomaly detection approach for directional drilling operations, called the analogues search model. It is designed for ranking the accidents from the knowledge base according to their relevance to the current situation in the drilling process, in order to find analogues and prevent anomalous behaviour. The solution is based on the compassion of mud logging data with the classification model built on Gradient Boosting of decision trees.

4.3.1 Related works

Anomaly detection is an important issue that has been investigated in various research areas: there are some examples of anomaly detection in Information Technology systems [103], medicine [104] and industry [105]. In oil and gas industry anomaly detection is widely spread: in downstream it is used for controlling pumping and pressure in different

systems, drilling process [106], lithology classification [99, 107]; in upstream, for example, engineers usually use it for detection sensors faults in the refinery [108] and pipelines [109].

Solving the problem of unusual behavior detection in drilling by analogues search, it is necessary to consider not only previous studies on time-series comparison and general algorithms of anomalies detection but also the methods and approaches for accidents detection during drilling, since they often happen as a result of anomalies.

4.3.1.1 Methods for time-series comparison

Considering the problem of analogues search, it is necessary to compare different time-series. Several authors [110–112] suggest measuring the similarity between two time-series by different metrics, for example, general Euclidean distance, Fourier coefficients, the Time-wrapping (TW) distance, and its modifications.

After the introduction of any distance, the whole database of time-series can be split into several groups with different clustering techniques, for example, K-means algorithm [113], mean-shift clustering [114], agglomerative hierarchical clustering [115]. Authors highlighted, that general Euclidean distance and Fourier coefficients showed themselves inefficient for time series with different length, while the cost for Time-wrapping distance computation for m-dimensional time-series might be significant. In this case, the mud logging patterns for different accidents and different oilfields are too diverse to apply such metrics effectively. Thus, clustering the raw time-series seems incompetent for the analogues search problem, which makes us move to a supervised approach for similarity learning based on statistical features extracted from time-series.

4.3.1.2 Methods for anomaly detection

Most of the general methods for anomaly detection were described previously, for example, in papers [103, 116, 117]. The authors distinguished several groups, based on statistical methods, machine learning, and unsupervised approaches. Due to the high variability of general methods and a large number of a review papers on them, we will not focus on description and will show only a few examples, which are relevant to the analogues search problem.

In the paper [116] the authors describe the approach, based on sliding window technique, in which some parts of time series with width w is converted into a single target value y_i by some particular classifier. By this principle, the sequences of signals were classified for the whole time-series signal as an anomaly or non-anomaly target value. The main

advantage of this method is the possibility of applying different existing classification methods. For anomaly detection in drilling, such an approach allows us to convert the unsupervised approach into supervised one, but do not involve physics of the drilling, which is significant for drilling accidents detection problem.

Nowadays, there are a lot of cases of neural networks [118] applications for anomaly detection [119, 120]. For example, the authors in [121] used a neural network to hierarchically learn features from the sensor measurements of exhaust gas temperatures and used them as the input to a neural network classifier for performing combustor anomaly detection. As a training set, the authors used 13791 samples before the accident. In our case, this approach may be inefficient due to the small size of the training sample and inability automatically handle missing values, which usually occur in mud logging data.

In the article [103] the authors highlight such approaches as a deviation of normal behavior and statistical methods. For example, [122] collected the stable database of activities not leading to intrusions and then used it to analyse the current behavior of the system by its comparison with database modes by different statistics. Comparing this approach with the problem of drilling accidents detection, it can be noticed that this approach is almost impossible to be used, because, unlike the user system, each well and field is unique. Usually, a similar slight deviation of normal drilling regime in one well can lead to serious accidents on the other.

4.3.1.3 Physics-based methods for drilling accident detection

The physics-based methods for detection accidents are primarily based on the monitoring and analysis of the key indicators of the drilling system. For example, [123] describes physical indicators and their changes, leading to failures. One of the main indicators of fluid shows while drilling is an increase in the volume of the drilling mud in the receiving tanks, reduction of standpipe pressure, an increase in the effluent flow rate with a constant flow of pumps, an unexpected increase in the mechanical penetration rate (due to a decrease in the density of the drilling mud, and, consequently, the pressure in the well). In case of wash-outs, the main evidence of the accident is in an increase of drill string weight, friction reduction, decrease in the fluid volume versus the calculated one while lifting the pipe string, movement of the drilling mud along the ditch system with the stopped circulation.

One more example of a physical-based method for failure detection is vibration, namely modeling the movement of the drill string and its components, which is represented in paper [124]. Early models of drill string dynamics have been developed primarily as an aid to the drilling engineers and rig designers, to help them understand wells behavior

and provide recommendations for improving the drilling operations. Currently, models are being used and investigated based on three parallel but different vibration modes, those help engineers detect anomaly by high vibration values.

Due to the inability to track all the indicators above, such physical-based methods are not suitable for solving our problem.

In addition to the methods shown above, there are different anomaly and failure patterns in mud logging plots. For example, a high number of drags and slack offs is used as the signs of a possible pipe stuck. The column drags usually occur when the column is lifted along with the increasing in hook load over its weight on the pipes. The slack off of the tool results in significant reduction in the load on the hook. Some evidence of columns stuck can be a stop of the column movement. In case of wash-outs, a decrease in pressure at a constant flow rate might be observed ([125]). The main failure pattern characterising the mud loss is a decrease in the volume in the tanks. The breakdown of the tools is marked by the pressure reduction during the constant flow rate simultaneously with a sharp drop in weight. So, particular patterns for each type of accidents on mud logs might be used as the first signs by which the model can determine the presence of failure.

4.3.2 Data overview

To solve the problem of failures detection by analogues search approach, a database with different types of accidents and their mud log data was collected.

Most of the failures happened in North and West Siberia oilfields and were composed of accident lessons that contain the information about these events: the exact date-time or depth at which the failure occurred. Such criterion was chosen in order to match the mud log data with the accident from the database and get a part of it that includes the failure. Each lesson included in the database also contained information about its accident type (stucks, wash-outs, breaks of drill pipe, mud loss, shale collars, gas, and water shows) and drilling operation at the moment of failure (tripping in, tripping out, drilling, cleaning, reaming). Such groups of accidents and drilling operations were chosen by the number of available cases and a possibility to be distinguished visually on mud logs.

In total, the database contains 94 lessons from 80 different wells and 19 oilfields. The summary of the size of different considered groups of accidents and related drilling operations is provided in Table 4.3.

	Triping in	Tripping out	Drilling	Cleaning	Reaming	Total
Stuck	18	11	10	0	1	40
Wash-outs	1	1	10	1	0	13
Breaks of drilling	1	2	4	6	0	13
Mud loss	2	2	6	0	1	11
Shale collars	0	0	9	0	0	9
Fluid shows	0	3	5	0	0	8
Total	22	19	44	7	2	94

TABLE 4.3: Breakdown of included accidents by type of accident and phase of drilling: in some cells we have almost no example for training

The considered measurements while drilling (MWD) data included the depth of the drill bit, torque on the rotor, weight on the hook, input pressure, rotation speed, a volume of input flow, a depth of the bottom hole, gas content, and weight on the bit.

4.3.3 Design of the analogues search model

In section 4.3.1, we discussed existing approaches for time-series comparison, anomaly, and failure detection. It was concluded, that for the problem of a drilling accident detection, it is necessary to use a supervised machine learning approach. The algorithm should take into account the particular mud logs pattern for different accident groups and be able to work with a small training set and corrupted or missing signal values.

We decided to solve the analogues search problem based on two-class classification of MWD pairs: for a specific well part, we need to understand whether something similar is present in the database by comparing features from MWD data of this part with those of entries in the database. Thus, there are two classes that determine whether two parts are similar or not.

For the current approach, we decided to build a classification model based on Gradient boosting of decision trees, because they are relatively undemanding in terms of sample size and data quality, can work with missing data, and learn quickly with a large number of features, what was shown in papers [126, 127].

The general principle of an analogues search model is shown in Figure 4.10. In order to take into account different patterns, for real-time signal and lessons from the database values of mean, variance, slope angle, absolute deviations, and relative coefficients of MWD time-series were calculated with different window sizes and were used as *input features* for Gradient boosting classification model. To assign targets, we assumed that pairs of intervals were similar if their accident types and drilling operations coincided. Henceforth we will refer to them as to *ground truth*

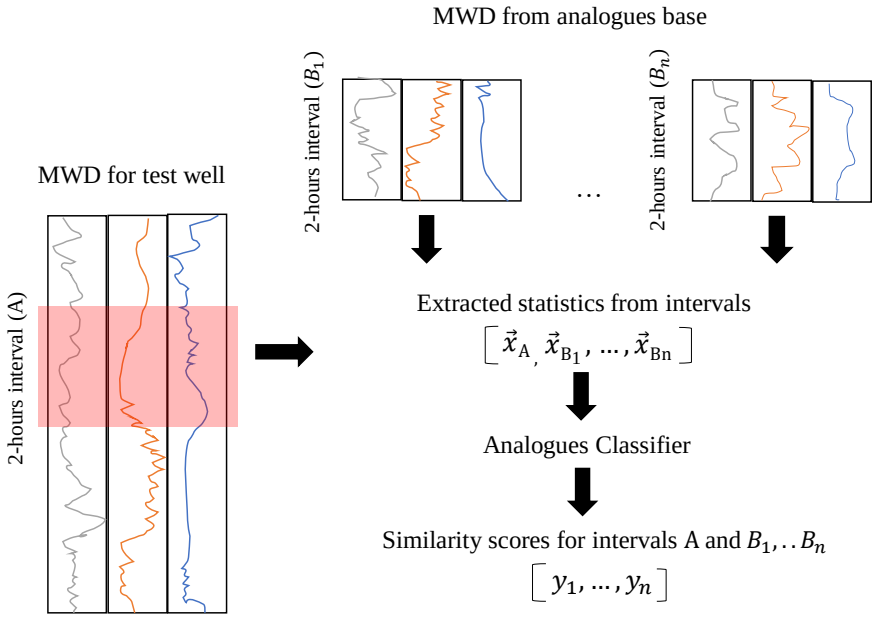


FIGURE 4.10: General scheme of analogues search model. Using 2 hours parts of MWD signals, different aggregated statistics were calculated. These features are inputs for gradient boosting classifier, that provides similarity scores for a pair of input signals.

To test the analogues model, several experiments were conducted. In order to validate our model, the standard quality metrics for the binary classification problem using leave-one-out cross-validation control were calculated. We also carried out a clustering analysis based on similarity values from the model to validate the aggregated statistics approach and evaluate the consistency of similarity learning. The similarity distributions between MWD data with accidents and random MWD parts of wells without abnormal behavior were analysed, in order to assess the model ability to distinguish regular drilling regime and accidents. In addition, we provided a sensitivity analysis with respect to various kinds of noise in MWD data.

4.3.4 Multi-fidelity active search for dataset annotation.

When a new accident case is being added to the dataset, an expert has to annotate analogous cases from the dataset to the new one. The amount of annotations needed to add a new case grows linearly with the dataset size, rapidly resulting in a tiresome routine as the dataset grows. In this section we show how this process can be facilitated by performing multi-fidelity active search of analogous accidents: low-fidelity relevance is a similarity predicted by the existing model, high-fidelity source is a label approved by an expert. To form a kernel matrix required for the MF-ASC model (see. chapter 3), we took differences between corresponding statistics extracted from time-series intervals

and calculated L2 norm of difference vectors (each component was normalized across all the vectors to have unit variance). The similarity between intervals, then, was obtained via Radial Basis Function transformation of the distances.

4.3.5 Results and discussions

4.3.5.1 Quality of the analogues search model

For the analogues search model, the cross-validation was carried out as follows:

1. For each of k iterations of cross-validation, random indexes of accidents from the database were generated for training and testing sets. The sets of wells for accidents were different for the train and test parts of the split.
2. The model was trained based on lessons, which indices were chosen as training ones.
3. Similarity values among entries in the training and test set were calculated. The model finds the analogue and, consequently, detects the failure, if the similarity value is bigger than the selected threshold ($s = 0.7$).
4. The predicted values were compared with ground truth labels.

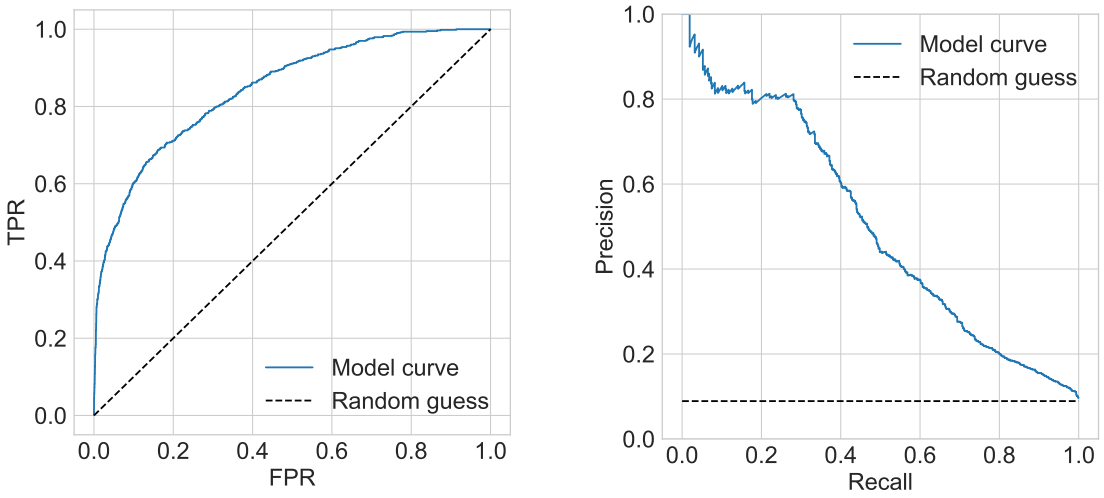
The results of cross-validation for the analogues search model are in Table 4.4. Using the current model, it is possible to identify almost all wells with abnormal drilling regime with low false alarm rate. So, it can be concluded that the model distinguishes different pairs quite well and identifies most of the similar ones.

	Predicted = 1	Predicted = 0
True = 1	5792	294
True = 0	223	345

Table 4.4: Confusion matrix for threshold $s = 0.7$

In order to obtain the model quality, two common metrics for classification problems were used: the area under the receiver operating characteristic curve (ROC AUC) and the area under the Precision-recall curve (PR AUC). The receiver operating characteristic (ROC) curve is presented in Figure 4.11. The area under ROC curve is 0.908, and significantly higher, than the area under the random guess classifier ROC curve 0.5. Since it is an unbalanced classification problem, a more suitable measure of model quality will be a

Precision-Recall curve, which is shown in Figure 4.11. The area under the Precision-Recall curve is 0.6086, which also indicates adequate model quality.



(A) ROC curve

(B) Precision Recall curve

FIGURE 4.11: Quality metrics for analogues search model. ROC AUC is 0.908, thus the model is significantly better than a random guess with ROC AUC 0.5. The area under PR curve is 0.6086, which is significantly better than the area under curve for a random guess approach 0.1.

4.3.5.2 Hold-out validation and threshold selection by analysis of confusion matrix

In this section, we used the analogues search model differently: it was applied to hold-out wells in order to understand how it works "in the wild." The analogues model was run on MWD signals from 30 hold-out cases, which included both normal and anomaly drilling modes.

Next, the threshold to balance the number of correct ($\mathcal{TP}$) and false ($\mathcal{FP}$) alarms was selected. After that, based on the true accident time for each well, the true and false model alarm rates were calculated as follows. We assumed, that the accident was correctly detected, and for this accident $\mathcal{TP} = 1$, if the similarity value was more than the chosen threshold, the model alarm was in the 4-hour interval before and 2 hours after the true accident ($\mathcal{TP}$ interval), and the most common accident type for the top-5 analogues matched with the true one.

In case of the false alarm, it was supposed that $\mathcal{FP} = 1$ for this interval if the model alarm was out of $\mathcal{TP}$ interval and there were no other alarms during the last hour. So if two or more alarms happened within 1 hour, it was counted as one false alarm. Here we also assumed, that predicted accident type was the most common one within top-5 analogues types; otherwise, it is supposed that $\mathcal{FP} = 1$.

To select the threshold, the total number of $\mathcal{TP}$ and $\mathcal{FP}$ for different threshold values was counted (Figure 4.12). For the threshold value 0.7, the total number of the model false alarms is less than 16 alarms per well, while the number of correct the alarms is still high.

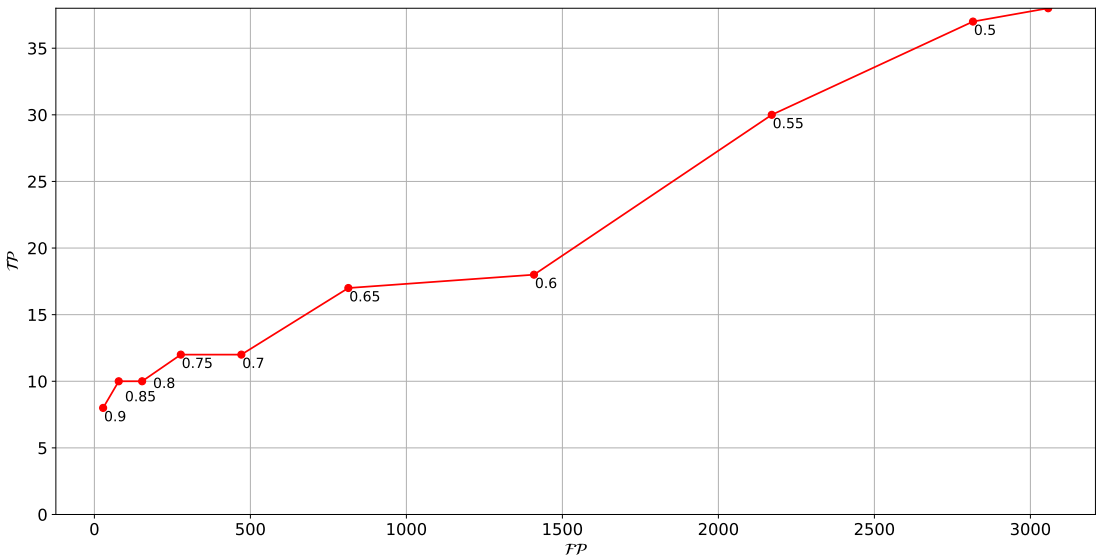


FIGURE 4.12: Total number of model correct (True Positive, TP) and false (False Positive, FP) alarms for different thresholds. Numbers at the curve are thresholds. The threshold 0.7 was used on final model.

Let us consider one example of analogues search model results for a hold-out well. In this case, the model ran on a well that contained a wash-out drilling accident. In this case, the model found an analogue with the same type of operation (drilling) as in the hold-out MWD measurements. It can be seen based on the general similarity of trends in such parameters as a hook position, depth of the drill bit, and bottom hole depth. In the cases, when similarity values exceeded the selected threshold, the model assumed, that current two hours in the past are similar to the analogue measurements. For both cases, we observe a decrease in pressure at a constant flow rate, which indicates the wash-out of the drill pipe. A careful examination of this case proves that the model correctly detected a wash-out accident and found an analogue. Similarity values and MWD signals for the "current" measurements and the analogue are presented in Figure 4.13 for one hold-out well.

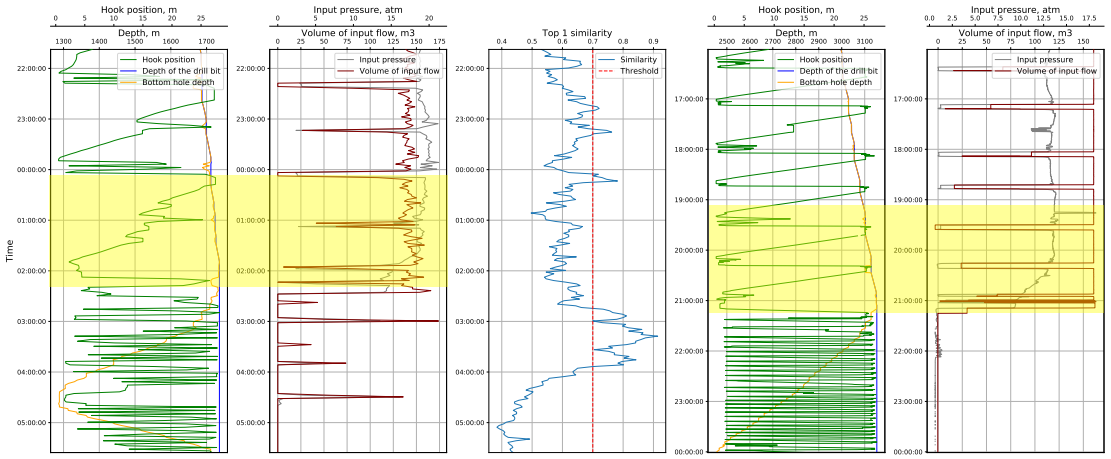


FIGURE 4.13: Analogues search model results in acion for hold-out well. Two plots on the right are MWD signals for a hold-out well, two plots on the left are the MWD for the analogue accident identified by the model. In the center there are similarity scores provided by the model: when similarity value exceeds the selected threshold the model alerts that the past two hours are similar to analogue measurements. For example, in the figure one of the similar areas for the hold-out case and analogue are highlighted in yellow colour. Both areas have similar signal trends, indicating a wash-out accident, which gives us an idea that the model correctly detected an accident and found a past analogue for it.

4.3.5.3 Clustering analysis

The dendrograms clustering analysis [128] was also used to assess the consistency of similarity learning. First of all, we represented it via adjacency matrix clusters based on the ground truth distribution of similarity. As mentioned earlier, two lessons are similar if their accident types and drilling operations are equal. Next, to compare initial distribution, similarity values that were used as an input parameter for constructing dendrograms were calculated in different ways:

- *Unsupervised comparison:* similarity values for lessons from the database were calculated only by the weighed l_1 norm among all MWD parameters, excluding the depth of the bottom hole and drill bit.
- *Using Gradient boosting technique:* dendrograms used similarities, that were calculated for the lessons from the training set, and resulted from the Gradient boosting model with aggregated statistics. This is an optimistic estimate of the quality of the similarity evaluations.
- *Cross-validation:* calculations were made with the model described in the previous step (using Gradient boosting technique) and cross-validation, which allows us to see how well the model generalises to the new cases of accidents. This is a more realistic estimate of the quality of the similarity evaluations.

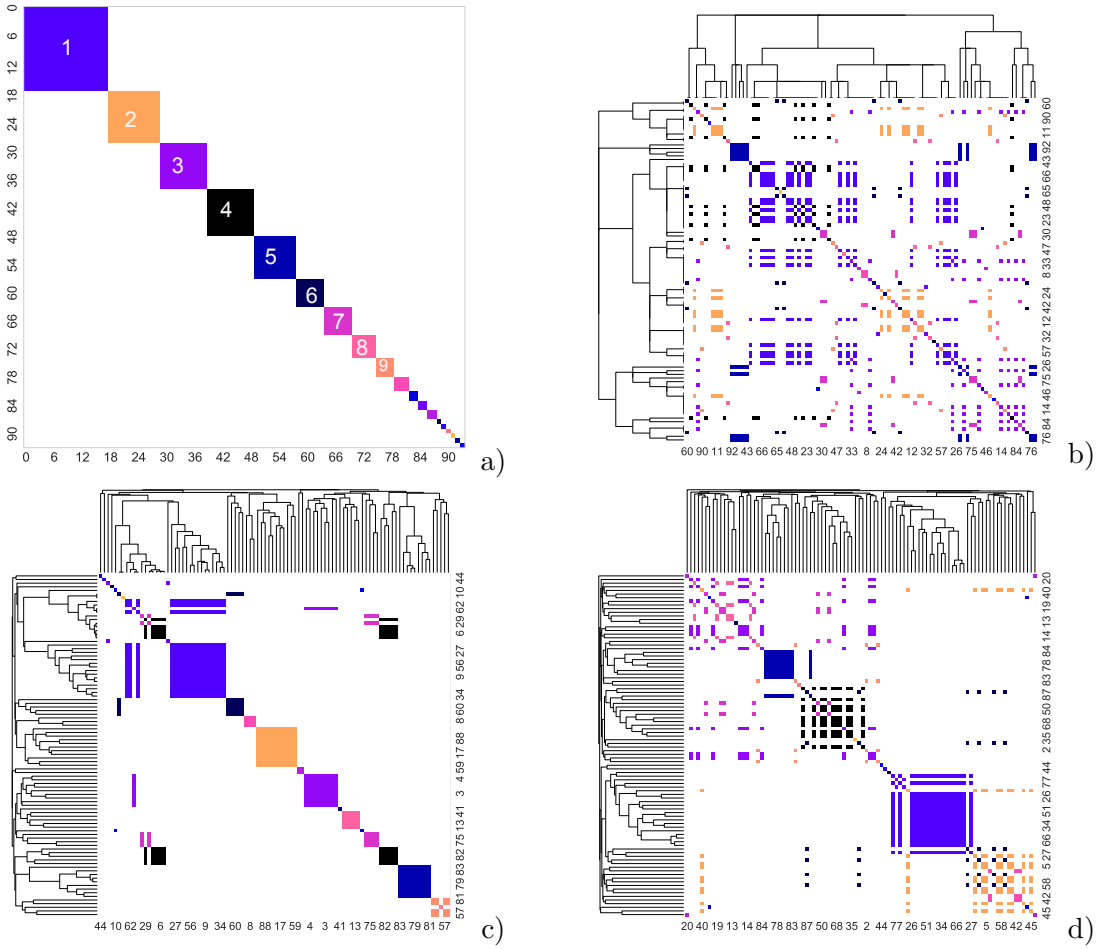


FIGURE 4.14: Clustering analysis: a) Initial clusters distribution, b) Dendrogram, based on simple comparison of MWD data, c) clusters, obtained from aggregated statistics and gradient boosting technique, d) dendrogram, obtained from cross-validation. The presence of colour at the intersection of row i and column j means that two cases,i and j respectively, of drilling accidents from the database belong to the same true accident group.

The results of the conducted test are shown in Figure 4.14 – clustering analysis represented by dendrograms, which illustrates how each cluster is composed by drawing a link between clusters: the top of the link indicates a cluster merge, while the two legs indicate which clusters were merged. In our case, each true group of accidents correspond to one colour, and is formed by the same type of drilling accidents, which have the same operation type. Since the main aim of this experiment is to see if similar anomalies can be grouped into clusters using different approaches, instead of the global order of clusters it is important to observe local proximity of the elements. Using aggregated statistics as an input for Gradient boosting classifier, the clusters distribution is more similar to the initial one, and it is possible to obtain more separated clusters, than by using only raw signals. The results obtained from cross-validation show us clearly selected clusters corresponding to different types of accidents and drilling operations. For some types of accident, the training set is quite complete, that can be seen by the trees above and to

the left of the plot, and has enough cases (clusters number 1,3,4,5). For others (clusters number 2,6-9), there is a greater distance for objects within the cluster. In our opinion, the reason for this might be the lack of examples of accidents in these groups. Consequently, for the correct determination of such groups of failures, the inclusion of a larger amount of data is required.

4.3.5.4 Robustness of the analogues search model

The analogues search model should meet the following two requirements. If an example from the training sample was submitted, the model should recognise it and provide it as the analogue with high similarity. Moreover, after reasonable distortion of such an example, the model should still recognise it.

While testing the first property is straightforward, to test the second one-two types of transformations were applied to the original time series: slight smoothing, distortion, and shift of data on the given number of time ticks (1 tick = 10 seconds). An example of the original and distorted time series for different values of the standard deviation is given in Fig 4.15.

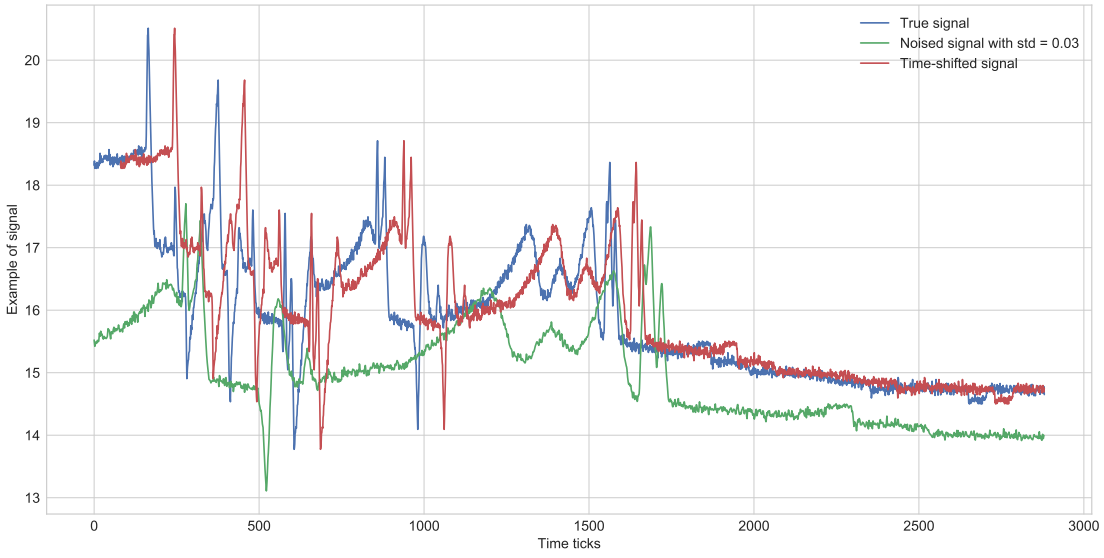


FIGURE 4.15: An example of original and distorted time series values

To understand how well the model distinguishes MWD parts of wells with normal behavior and ones with accidents, the model was trained on MWD parts, corresponding to the lessons from the current database and tested for normal and distorted parts.

The distribution of obtained similarity values was presented as box-plots for different testing sets: random parts without accidents, intervals with accidents, time-shifted duplicates of the intervals with accidents and copies of the intervals with accidents with

varying levels of shift and noise. The distortion of original time-series was done by the multiplication of a smooth curve with average mean 1 and the given standard deviation.

The numerical characteristic R was also calculated: the difference between the 90% quantile of the random parts set and the 10% quantile of the data that we would like to highlight. Valid values are bigger than 0; good ones are more than 0.2. Standard deviations for R were calculated using the bootstrap technique [129], the calculation used 100 samples.

The box-plots for the cases, mentioned above, and values of R coefficient, which characterises the difference in the similarity values for two different sets of intervals, are in Figure 4.16.

The analogues search model shows high similarity values for noised lessons with standard deviations as high as 0.01 and time-shifted lessons as high as 20 ticks. So, in these cases, the model finds similar sections from the training set.

At the same time, the similarity values for normal parts are low, which shows the model ability to distinguish a normal drilling mode from the accidents-related drilling mode. It also can be seen that the model can separate random MWD parts from the data, corresponding to the lessons from the database, for shifting up to 400 seconds and for noise with a standard deviation of up to 0.03.

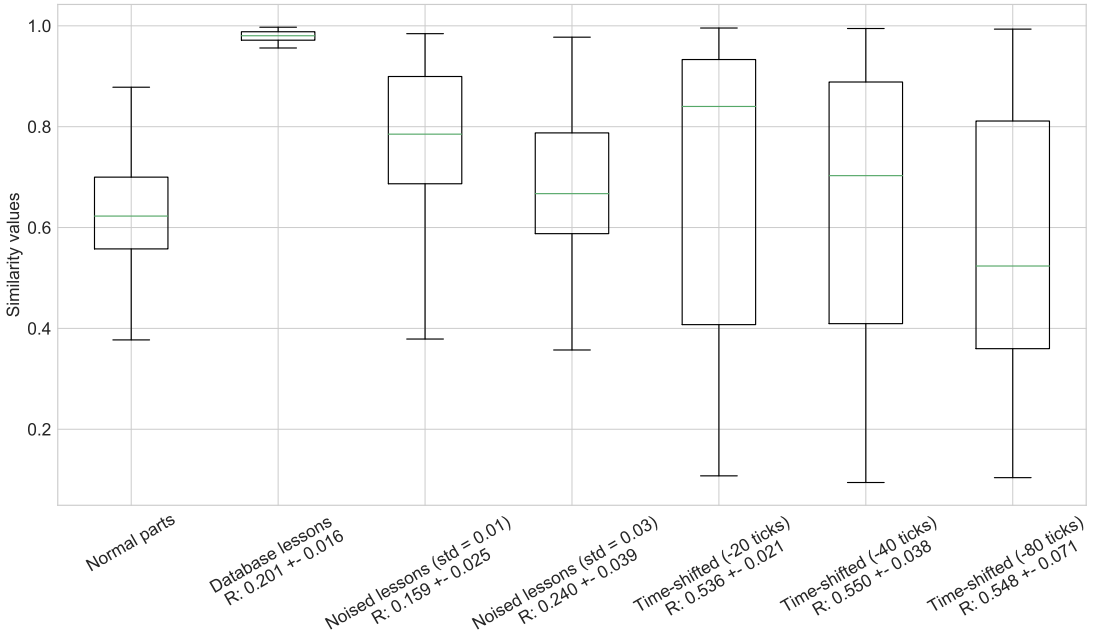


FIGURE 4.16: Box-plots for different intervals. Such figure gives us an idea of how much MWD data can be distorted, so that the model can still recognise them

4.3.5.5 Annotation of new cases with MF-ASC

In this section we report the experiments on application of multi-fidelity active search for dataset annotation introduced in section 4.3.4.

The application of this method makes sense only for classes of accidents that are well-represented in the dataset; thus, we test active search for the types with at least five cases in the dataset. For each case, we run 10 search sessions with different initially annotated sets, that include three high-fidelity entries and three low-fidelity ones.

To quantify the performance of the annotation procedure, we measure the share of the found objects of the same accident class, reviewed by an expert with respect to the total number of reviews. The results are shown in figure 4.17. This figure demonstrates that MF-ASC can find the majority (around 80 percent) of analogues twice faster than the random search, so on average our method reduces experts efforts twofold.

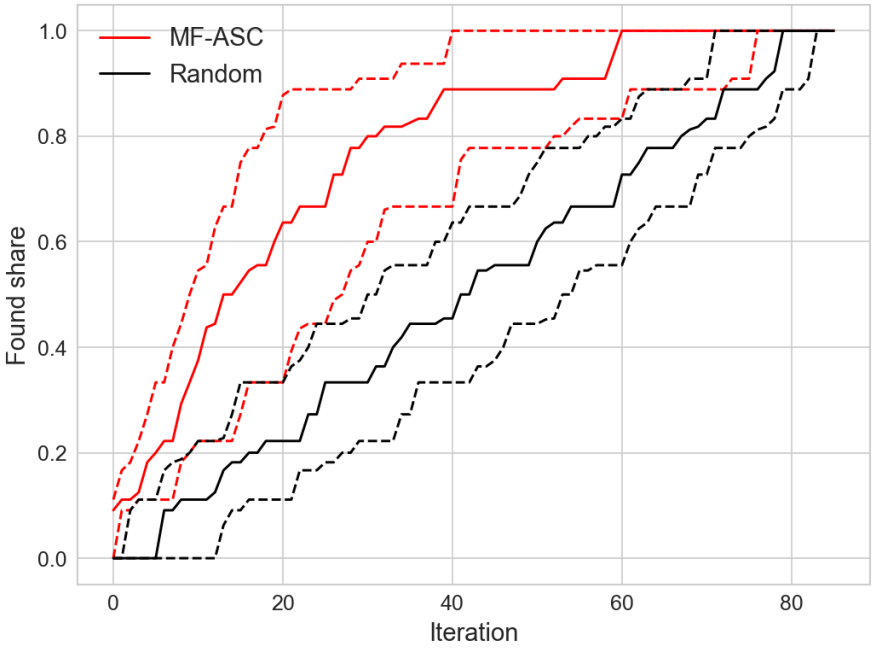


FIGURE 4.17: Multi-fidelity active search performance compared to random search of analogues. Bold curves correspond to median and dashed curves correspond to 0.2- and 0.8-quantiles over multiple experiments.

4.3.5.6 Discussion

When a drilling engineer encounters an accident, he or she can identify the type of the accident and rely on the actions taken during the similar cases in the past. Moreover, from now, it is rather evident, that it is possible to correlate the real-time state of the

drilling process with the past events from the historical database with machine learning (see sections 4.3.5.1, 4.3.5.3 and 4.3.5.4).

The quality of the machine learning model is acceptable for use during drilling as even a single successful application of the developed model can save time and money. However, there is still room for improvement. In particular, the model quality for identification of the accidents types underrepresented in the database is sometimes low. Thus, to maximize quality, a larger database is required. Replenishment of the database can be performed using the multi-fidelity active search method as follows: since the analogues search algorithm produces False Positive errors, experts can review some of them in more detail, because these cases could actually contain missing accidents in the database or situations that were nearly accidental. The feedback from the experts will be gradually utilized by MF-ASC to yield more insistent results.

In the separate section, we address the issue of model reliability and generalization ability. The conducted tests suggest a limit of applicability of the model, as after significant distortion of the initial signal the model no longer works correctly. However, the obtained results correspond to the general machine learning theory of extrapolation and generalization properties of the data-driven models.

4.3.6 Conclusions

We developed a real-time analogues search model that detects anomalies and finds analogues in a database of historical data.

The anomaly detection is based on the smart comparison of the real-time MWD data and the MWD data from the historical database followed by ranking the lessons from the database by their similarity to the real-time state. The comparison and ranking utilize Gradient boosting classification model.

The conducted analysis of the analogues search model showed that the obtained quality metrics, such as ROC AUC (0.908) and PR AUC (0.6086), are significantly higher than the same metrics for the random guess classifier (ROC AUC: 0.5, PR AUC: 0.1). The obtained metrics suggest that the model is of reasonable quality and can distinguish pairs of similar and non-similar cases well.

The clustering analysis showed that the use of basic MWD signals is not sufficient for the selection of analogues, and in general for the analogues search model. We have discovered that the introduction of aggregated statistics as input for Gradient boosting classifier allows finding a sufficiently larger number of analogues of real-time signals.

According to the robustness analysis, the model identifies the lessons from the training sample, if such lessons are also in a testing sample. The analogue search quality remains high even after reasonable distortion of the examples from the training sample by adding noise and shifts to the initial signal. These experiments helped to identify the limits of applicability of the model, ensuring good understanding of what level of MWD signal distortion is still acceptable for an accurate analogues identification.

We also demonstarted that the dataset annotation process can be significantly facilitated using a multi-fidelity active search tool. This method will be especially helpful for further replenishment of the dataset, as it is advised in the discussion.

Chapter 5

Conclusion

Modern machine-learning projects usually involve the data from various sources, some of which can be precise, but typically much more expensive than others, that are approximate. Multi-fidelity models gain interest as they promise to meet the desired quality with less resources, that a single-fidelity model would require. These models are especially useful for several rapidly developing areas, such as Auto-ML [130], where training with less data points or less epochs can serve as a low-fidelity cheap approximation of the final model's validation performance; and large dataset annotation [131], where annotators with various levels of expertise and price can be employed to achieve cost-effective results. Active search is also going to play an essential role with the spread of the voice assistants [132] and other augmented intelligence technologies. Unlike traditional work with search engines, where many results can be displayed at a time, queries to voice assistants typically return only one object, therefore it is especially important to adapt to user preferences on the fly to minimize the amount of interactions.

In this thesis, we studied multi-fidelity classification and active search methods and through them contributed to the broader research fields such as Bayesian optimization, surrogate multi-fidelity modeling and machine learning in the following ways:

In chapter 2, a new MF gpc method was proposed for modeling the relevance of objects from heterogeneous data sources using a co-kriging scheme on the latent Gaussian processes for the classification problem. For this method, generalization from one to multiple data sources of approximate Bayesian inference using the Laplace approximation was developed to provide computationally effective predictions. We have experimentally demonstrated that MF gpc was more resistant to different noise levels in low-fidelity data, and also investigated under what conditions of the noise level, adding low-fidelity data to the training sample improves the quality of MF gpc relative to the usual classification with only one source with high fidelity data.

In chapter 3, a novel algorithm MF-ASC has been developed for active search of objects in the presence of heterogeneous data sources using a regression based on the Gaussian processes and co-kriging scheme. The characteristics of the algorithm, such as sensitivity to the correlation between sources, granularity of the feedback, scalability of computations in time, and others, were experimentally investigated.

Apart theoretic and algorithmic contributions, in chapter 4 the developed methodology and instrumentarium found a use in several industrial applications: the utility of Bayesian optimization in general was demonstrated in the active muon optimization for the SHiP Shield experiment at CERN; MF gpc was applied for refining the rock type labels in the project on the data-driven model for lithotype identification at directional oil well drilling; MF-ASC was also useful to reduce the manual labor for annotating similar cases of drilling accidents in the project on a data-driven accidents detection during directional oil well drilling. Furthermore, the method MF-ASC has recently been integrated into Vega[1] [133] library for Neural Architecture Search.

[1] https://github.com/huawei-noah/vega

Appendix A

Supplementary materials

This appendix contains supplementary materials for experimental results.

TABLE A.1: Average ROC AUC among multiple runs on artificial datasets from group 1. Margins indicate standard deviations of mean estimates.

Noise level	0.2				0.4			
Dimensionality	2D	5D	10D	20D	2D	5D	10D	20D
MF gpc	**0.975 ± 0.003**	**0.853 ± 0.005**	**0.716 ± 0.015**	**0.643 ± 0.012**	**0.968 ± 0.005**	**0.750 ± 0.010**	**0.615 ± 0.013**	0.573 ± 0.011
gpc	**0.970 ± 0.005**	0.732 ± 0.012	0.616 ± 0.012	0.587 ± 0.006	**0.970 ± 0.005**	**0.732 ± 0.012**	**0.616 ± 0.012**	**0.587 ± 0.006**
logit	0.738 ± 0.026	0.590 ± 0.012	0.559 ± 0.008	0.559 ± 0.007	0.738 ± 0.026	0.590 ± 0.012	0.559 ± 0.008	0.559 ± 0.007
xgb	0.914 ± 0.014	0.662 ± 0.012	0.591 ± 0.007	0.574 ± 0.006	0.914 ± 0.014	0.662 ± 0.012	0.591 ± 0.007	0.574 ± 0.006
C gpc	0.944 ± 0.005	**0.854 ± 0.005**	**0.721 ± 0.015**	**0.654 ± 0.008**	0.811 ± 0.010	0.683 ± 0.015	**0.626 ± 0.011**	**0.592 ± 0.006**
C logit	0.721 ± 0.030	0.619 ± 0.010	0.580 ± 0.007	0.585 ± 0.005	0.675 ± 0.036	0.584 ± 0.008	0.557 ± 0.008	0.557 ± 0.007
C xgb	0.916 ± 0.011	0.725 ± 0.009	0.644 ± 0.009	0.607 ± 0.005	0.807 ± 0.015	0.637 ± 0.010	0.586 ± 0.009	0.567 ± 0.005
S gpc	0.949 ± 0.007	0.812 ± 0.007	0.686 ± 0.011	0.616 ± 0.007	0.938 ± 0.009	0.713 ± 0.011	**0.617 ± 0.010**	**0.589 ± 0.007**
S logit	0.740 ± 0.027	0.592 ± 0.012	0.563 ± 0.007	0.559 ± 0.007	0.742 ± 0.026	0.591 ± 0.012	0.561 ± 0.008	0.559 ± 0.007
S xgb	0.921 ± 0.011	0.700 ± 0.010	0.608 ± 0.009	0.583 ± 0.006	0.914 ± 0.014	0.657 ± 0.011	0.591 ± 0.007	0.575 ± 0.006
hetmogp	0.909 ± 0.016	0.500 ± 0.000	0.500 ± 0.000	0.500 ± 0.000	0.802 ± 0.021	0.500 ± 0.000	0.500 ± 0.000	0.500 ± 0.000

TABLE A.2: Average ROC AUC among multiple runs on datasets from group 2 with noise level 0.2. Margins indicate standard deviations of mean estimates.

	dbts	grmn	stmg	mshr	splc	spmb	hpth	wvfr
MF gpc	**0.805 ± 0.006**	0.702 ± 0.008	**0.997 ± 0.000**	**0.997 ± 0.002**	0.936 ± 0.002	0.925 ± 0.006	0.646 ± 0.009	**0.919 ± 0.008**
gpc	0.778 ± 0.009	0.704 ± 0.012	0.997 ± 0.000	**0.995 ± 0.002**	0.901 ± 0.013	0.907 ± 0.010	0.633 ± 0.018	**0.908 ± 0.008**
logit	**0.812 ± 0.009**	0.683 ± 0.028	**0.998 ± 0.000**	**0.994 ± 0.002**	0.913 ± 0.013	0.915 ± 0.007	**0.772 ± 0.031**	0.858 ± 0.010
xgb	0.742 ± 0.007	**0.702 ± 0.027**	0.982 ± 0.007	0.987 ± 0.003	**0.971 ± 0.004**	0.925 ± 0.003	**0.827 ± 0.054**	0.886 ± 0.003
C gpc	**0.804 ± 0.005**	0.699 ± 0.007	0.996 ± 0.001	**0.995 ± 0.004**	0.937 ± 0.002	0.914 ± 0.012	0.570 ± 0.023	**0.910 ± 0.011**
C logit	**0.803 ± 0.009**	**0.704 ± 0.014**	0.989 ± 0.002	0.955 ± 0.009	0.794 ± 0.013	0.859 ± 0.025	0.654 ± 0.019	0.820 ± 0.025
C xgb	0.767 ± 0.009	0.696 ± 0.011	0.987 ± 0.001	0.987 ± 0.003	0.958 ± 0.008	**0.946 ± 0.006**	**0.791 ± 0.050**	0.891 ± 0.014
S gpc	**0.804 ± 0.005**	**0.725 ± 0.009**	**0.997 ± 0.000**	**0.997 ± 0.001**	0.915 ± 0.012	0.914 ± 0.005	0.616 ± 0.018	**0.918 ± 0.009**
S logit	**0.812 ± 0.010**	0.684 ± 0.027	0.997 ± 0.000	**0.994 ± 0.002**	0.924 ± 0.010	0.923 ± 0.006	**0.766 ± 0.037**	0.861 ± 0.012
S xgb	0.738 ± 0.005	0.687 ± 0.017	0.971 ± 0.012	0.983 ± 0.004	**0.967 ± 0.010**	**0.943 ± 0.004**	**0.766 ± 0.031**	0.895 ± 0.006
hetmogp	0.500 ± 0.000	0.500 ± 0.000	0.500 ± 0.000	0.500 ± 0.000	0.500 ± 0.000	0.500 ± 0.000	0.500 ± 0.000	0.500 ± 0.000

TABLE A.3: Average ROC AUC among multiple runs on datasets from group 2 with noise level 0.4. Margins indicate standard deviations of mean estimates.

	dbts	grmn	stmg	mshr	splc	spmb	hpth	wvfr
MF gpc	0.781 ± 0.009	**0.710 ± 0.010**	0.997 ± 0.000	**0.996 ± 0.002**	0.905 ± 0.011	0.914 ± 0.004	0.676 ± 0.017	**0.909 ± 0.009**
gpc	0.778 ± 0.009	**0.704 ± 0.012**	0.997 ± 0.000	**0.995 ± 0.002**	0.901 ± 0.013	0.907 ± 0.010	0.633 ± 0.018	**0.908 ± 0.008**
logit	**0.812 ± 0.009**	**0.683 ± 0.028**	**0.998 ± 0.000**	**0.994 ± 0.002**	0.913 ± 0.013	**0.915 ± 0.007**	**0.772 ± 0.031**	0.858 ± 0.010
xgb	0.742 ± 0.007	**0.702 ± 0.027**	0.982 ± 0.007	0.987 ± 0.003	**0.971 ± 0.004**	**0.925 ± 0.003**	**0.827 ± 0.054**	0.886 ± 0.003
C gpc	0.685 ± 0.093	0.642 ± 0.017	0.986 ± 0.003	0.981 ± 0.009	0.846 ± 0.019	0.852 ± 0.023	0.479 ± 0.011	0.840 ± 0.030
C logit	0.743 ± 0.018	0.630 ± 0.001	0.934 ± 0.009	0.827 ± 0.026	0.626 ± 0.002	0.725 ± 0.046	0.579 ± 0.026	0.711 ± 0.042
C xgb	0.697 ± 0.022	0.621 ± 0.014	0.934 ± 0.012	0.921 ± 0.024	0.831 ± 0.008	0.849 ± 0.038	0.674 ± 0.033	0.771 ± 0.043
S gpc	0.791 ± 0.003	**0.711 ± 0.010**	0.997 ± 0.000	**0.996 ± 0.002**	0.901 ± 0.013	0.906 ± 0.009	0.622 ± 0.017	**0.907 ± 0.010**
S logit	**0.811 ± 0.009**	**0.683 ± 0.029**	**0.997 ± 0.000**	**0.994 ± 0.002**	0.914 ± 0.013	**0.916 ± 0.007**	**0.771 ± 0.027**	0.858 ± 0.010
S xgb	0.747 ± 0.005	**0.680 ± 0.032**	0.984 ± 0.003	0.988 ± 0.001	**0.972 ± 0.005**	**0.927 ± 0.006**	**0.752 ± 0.035**	0.885 ± 0.003
hetmogp	0.500 ± 0.000	0.500 ± 0.000	0.500 ± 0.000	0.500 ± 0.000	0.500 ± 0.000	0.500 ± 0.000	0.500 ± 0.000	0.500 ± 0.000

Bibliography

[1] Carl Edward Rasmussen and Christopher KI Williams. *Gaussian processes for machine learning*. MIT Press, 2006.

[2] Benjamin Peherstorfer, Karen Willcox, and Max Gunzburger. Survey of multifidelity methods in uncertainty propagation, inference, and optimization. *Preprint*, pages 1–57, 2016.

[3] Burr Settles. *Active Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2012.

[4] A Akmete, A Alexandrov, A Anokhina, Shuji Aoki, Erica Atkin, N Azorskiy, JJ Back, A Bagulya, A Baranov, GJ Barker, et al. The active muon shield in the ship experiment. *Journal of Instrumentation*, 12:P05011, 2017.

[5] Hannes Nickisch and Carl Edward Rasmussen. Approximations for binary Gaussian process classification. *Journal of Machine Learning Research*, 9(Oct):2035–2078, 2008.

[6] Edward Snelson and Zoubin Ghahramani. Sparse gaussian processes using pseudo-inputs. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems*, volume 18, pages 1257–1264. MIT Press, 2006.

[7] Chuong B. Do. More on multivariate gaussians, 2008.

[8] Jochen Görtler, Rebecca Kehlbeck, and Oliver Deussen. A visual exploration of gaussian processes. *Distill*, 2019. doi: 10.23915/distill.00017. https://distill.pub/2019/visual-exploration-gaussian-processes.

[9] M. Giselle Fernández-Godino, Chanyoung Park, Nam-Ho Kim, and Raphael T. Haftka. Review of multi-fidelity models, 2016.

[10] David J. J. Toal. Some considerations regarding the use of multi-fidelity kriging in the construction of surrogate models. *Structural and Multidisciplinary Optimization*, 51(6):1223–1245, Jun 2015. ISSN 1615-1488. doi: 10.1007/s00158-014-1209-5. URL `https://doi.org/10.1007/s00158-014-1209-5`.

[11] Hao Zhang and Wenxiang Cai. When doesn't cokriging outperform kriging? *Statistical Science*, 30(2):176–180, 2015.

[12] E. Burnaev and A. Zaytsev. Minimax approach to variable fidelity data interpolation. *Proceedings of Machine Learning Research, Artificial Intelligence and Statistics*, 54:652–661, 2016.

[13] Roman Garnett, Yamuna Krishnamurthy, Xuehan Xiong, Jeff G. Schneider, and Richard P. Mann. Bayesian optimal active search and surveying. In *ICML*, page 843–850, 2012.

[14] Eric Brochu, Vlad M Cora, and Nando de Freitas. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. eprint arXiv:1012.2599, arXiv.org, December 2010.

[15] Peter I Frazier. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

[16] Hastagiri P. Vanchinathan, Andreas Marfurt, Charles-Antoine Robelin, Donald Kossmann, and Andreas Krause. Discovering valuable items from massive data. In *KDD*, pages 1195–1204, 2015.

[17] Alexey Zaytsev and Evgeny Burnaev. Minimax approach to variable fidelity data interpolation. In *AISTATS*, pages 652–661, 2017.

[18] David J. Toal. Some considerations regarding the use of multi-fidelity kriging in the construction of surrogate models. *Struct. Multidiscip. Optim.*, 51(6):1223–1245, June 2015. ISSN 1615-147X.

[19] Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y Ng. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 254–263. Association for Computational Linguistics, 2008.

[20] Witold Pawlus, Kjell Gunnar Robbersmyr, and Hamid Reza Karimi. Mathematical modeling and parameters estimation of a car crash using data-based regressive model approach. *Applied Mathematical Modelling*, 35(10):5091 – 5107, 2011.

[21] David H Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992.

[22] Xuhui Meng and George Em Karniadakis. A composite neural network that learns from multi-fidelity data: Application to function approximation and inverse pde problems. *Journal of Computational Physics*, 401:109020, 2020.

[23] Xuhui Meng, Hessam Babaee, and George Em Karniadakis. Multi-fidelity bayesian neural networks: Algorithms and applications. *Journal of Computational Physics*, 438:110361, 2021.

[24] Pablo Moreno-Muñoz, Antonio Artés-Rodríguez, and Mauricio A Álvarez. Heterogeneous multi-output gaussian process prediction. *arXiv preprint arXiv:1805.07633*, 2018.

[25] Evgenii Tsymbalov, S. Makarychev, A. Shapeev, and Maxim Panov. Deeper connections between neural networks and gaussian processes speed-up active learning. *ArXiv*, abs/1902.10350, 2019.

[26] D. Barber. *Bayesian Reasoning and Machine Learning*. Cambridge University Press, 2012.

[27] Alexander IJ Forrester, András Sóbester, and Andy J Keane. Multi-fidelity optimization via surrogate modelling. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 463(2088):3251–3269, 2007.

[28] Alexey Zaytsev and Evgeny Burnaev. Large scale variable fidelity surrogate modeling. *Annals of Mathematics and Artificial Intelligence*, 81(1):167–186, 2017.

[29] Christoph Dribusch. *Multi-fidelity construction of explicit boundaries: Application to aeroelasticity*. The University of Arizona, 2013.

[30] Jean-Baptiste Mouret and Konstantinos Chatzilygeroudis. 20 years of reality gap: a few thoughts about simulators in evolutionary robotics. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 1121–1124. ACM, 2017.

[31] Michalis Titsias. Variational learning of inducing variables in sparse gaussian processes. In *Artificial Intelligence and Statistics*, pages 567–574. PMLR, 2009.

[32] Rishit Shetha, Yuyang Wangb, COM Roni Khardona, and TUFTS EDU. Sparse variational inference for generalized gaussian process models. In *ICML*, pages 1301–1311, 2015.

[33] James Hensman, Alexander Matthews, and Zoubin Ghahramani. Scalable variational gaussian process classification. In *Artificial Intelligence and Statistics*, pages 351–360. PMLR, 2015.

[34] Daniel Hernández-Lobato and José Miguel Hernández-Lobato. Scalable gaussian process classification via expectation propagation. In *Artificial Intelligence and Statistics*, pages 168–176. PMLR, 2016.

[35] Ke Alexander Wang, Geoff Pleiss, Jacob R Gardner, Stephen Tyree, Kilian Q Weinberger, and Andrew Gordon Wilson. Exact gaussian processes on a million data points. *arXiv preprint arXiv:1903.08114*, 2019.

[36] Botond Cseke and Tom Heskes. Improving posterior marginal approximations in latent gaussian models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 121–128. JMLR Workshop and Conference Proceedings, 2010.

[37] Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. In *Advances in Neural Information Processing Systems*, pages 1196–1204, 2013.

[38] Chris Williams, Edwin V Bonilla, and Kian M Chai. Multi-task gaussian process prediction. *Advances in Neural Information Processing Systems*, pages 153–160, 2007.

[39] Yu Zhang and Qiang Yang. A survey on multi-task learning. *arXiv preprint arXiv:1707.08114*, 2017.

[40] Mauricio A Álvarez and Neil D Lawrence. Computationally efficient convolved multiple output gaussian processes. *Journal of Machine Learning Research*, 12(May):1459–1500, 2011.

[41] Marc C Kennedy and Anthony O'Hagan. Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, 87(1):1–13, 2000.

[42] Marc C Kennedy and Anthony O'Hagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3):425–464, 2001.

[43] Loic Le Gratiet. Multi-fidelity gaussian process regression for computer experiments, 2013.

[44] Loic Le Gratiet and Josselin Garnier. Recursive co-kriging model for design of computer experiments with multiple levels of fidelity. *International Journal for Uncertainty Quantification*, 4(5): 365–386, 2014.

[45] Emanuele Giorgi, Sanie S. S. Sesay, Dianne J. Terlouw, and Peter J. Diggle. Combining data from multiple spatially referenced prevalence surveys using generalized linear geostatistical models. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 178(2):445–464, 2015.

[46] Jarno Vanhatalo, Marcelo Hartmann, and Lari Veneranta. Joint species distribution modeling with additive multivariate gaussian process priors and heteregenous data. *arXiv preprint arXiv:1809.02432*, 2018.

[47] Alexander I. J. Forrester, Andrs Sbester, and Andy J. Keane. *Engineering Design via Surrogate Modelling*. John Wiley & Sons, Ltd, jul 2008. doi: 10.1002/9780470770801.

[48] Robert S. Liptser and Albert N. Shiryaev. *Statistics of Random Processes*. Springer Berlin Heidelberg, 2001. doi: 10.1007/978-3-662-10028-8.

[49] Christopher M Bishop. *Neural networks for pattern recognition*. Oxford university press, 1995.

[50] David JC MacKay. The evidence framework applied to classification networks. *Neural Computation*, 4(5):720–736, 1992.

[51] Christopher KI Williams and David Barber. Bayesian classification with Gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1342–1351, 1998.

[52] Kathrin Schäcke. On the Kronecker Product, 2013.

[53] James A Hanley and Barbara J McNeil. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36, 1982.

[54] Randal S Olson, William La Cava, Patryk Orzechowski, Ryan J Urbanowicz, and Jason H Moore. Pmlb: a large benchmark suite for machine learning evaluation and comparison. *BioData Mining*, 10(1):1–13, 2017.

[55] Filipe Rodrigues, Francisco Pereira, and Bernardete Ribeiro. Gaussian process classification and active learning with multiple annotators. In *International Conference on Machine Learning*, pages 433–441, 2014.

[56] Pablo Ruiz, Emre Besler, Rafael Molina, and Aggelos K Katsaggelos. Variational Gaussian process for missing label crowdsourcing classification problems. In *Machine Learning for Signal Processing (MLSP), 2016 IEEE 26th International Workshop on*, pages 1–6. IEEE, 2016.

[57] John Salvatier, Thomas V Wiecki, and Christopher Fonnesbeck. Probabilistic programming in python using pymc3. *PeerJ Computer Science*, 2:e55, 2016.

[58] Paul Resnick and Hal R. Varian. Recommender systems. *Commun. ACM*, 40(3):56–58, 1997. ISSN 0001-0782.

[59] Xuezhi Wang, Roman Garnett, and Jeff Schneider. Active search on graphs. In *KDD*, pages 731–738, 2013.

[60] Yifei Ma, Tzu-Kuo Huang, and Jeff G. Schneider. Active search and bandits on graphs using sigma-optimality. In *UAI*, pages 542–551, 2015.

[61] Apostolos N. Burnetas and Michael N. Katehakis. Optimal adaptive policies for markov decision processes. *Math. Operations Research*, 22(1):222–255, 1997.

[62] Alexander I. J. Forrester, András Sóbester, and Andy J. Keane. Multi-fidelity optimization via surrogate modelling. *Proc. R. Soc. A*, 463(2088):3251–3269, 2007.

[63] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*, volume 1. The MIT Press, 2006.

[64] Natalia M. Alexandrov, John E Dennis Jr, Robert Michael Lewis, and Virginia Torczon. A trust region framework for managing the use of approximation models in optimization. *Structural Optimization*, 15(1):16–23, 1998.

[65] Shawn E. Gano, John E. Renaud, and Brian Sanders. Hybrid variable fidelity optimization by using a kriging-based scaling function. *AIAA Journal*, 43:2422–2433, 11 2005.

[66] Simon Tong and Edward Chang. Support vector machine active learning for image retrieval. In *MULTIMEDIA*, pages 107–118, 2001.

[67] Kirthevasan Kandasamy, Gautam Dasarathy, Barnabas Poczos, and Jeff Schneider. The multi-fidelity multi-armed bandit. In *NIPS*, pages 1777–1785, 2016.

[68] Matthias Poloczek, Jialei Wang, and Peter Frazier. Multi-information source optimization. In *NIPS*, pages 4288–4298, 2017.

[69] Yehong Zhang, Trong Nghia Hoang, Bryan Kian Hsiang Low, and Mohan Kankanhalli. Information-based multi-fidelity bayesian optimization. In *NIPS Workshop on Bayesian Optimization*, 2017.

[70] Kirthevasan Kandasamy, Gautam Dasarathy, Junier B Oliva, Jeff Schneider, and Barnabás Póczos. Gaussian process bandit optimisation with multi-fidelity evaluations. In *NIPS*, pages 992–1000, 2016.

[71] Yoshiharu Ishikawa, Ravishankar Subramanya, and Christos Faloutsos. Mindreader: Querying databases through multiple examples. In *VLDB*, pages 218–227, 1998.

[72] Matthew Schultz and Thorsten Joachims. Learning a distance metric from relative comparisons. In *NIPS*, pages 41–48, 2003.

[73] Jason V. Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S. Dhillon. Information-theoretic metric learning. In *ICML*, pages 209–216, 2007.

[74] Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *WWW*, pages 661–670, 2010.

[75] Andrew March, Karen Willcox, and Qiqi Wang. Gradient-based multifidelity optimisation for aircraft design using bayesian model calibration. *The Aeronautical Journal*, 115(1174):729–738, 2011.

[76] Evgeny Burnaev and Maxim Panov. Adaptive design of experiments based on gaussian processes. In *International Symposium on Statistical Learning and Data Sciences*, pages 116–125. Springer, 2015.

[77] Remi R. Lam, Karen E. Willcox, and David H. Wolpert. Bayesian optimization with a finite budget: an approximate dynamic programming approach. In *NIPS*, pages 883–891, 2016.

[78] Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias W. Seeger. Information-theoretic regret bounds for gaussian process optimization in the bandit setting. *IEEE Trans. Information Theory*, 58(5):3250–3265, 2012.

[79] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[80] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *KDD*, pages 855–864, 2016.

[81] Gideon Dror, Noam Koenigstein, Yehuda Koren, and Markus Weimer. The yahoo! music dataset and kdd-cup '11. In *Proc. KDD Cup 2011*, pages 8–18, 2012.

[82] Mark E. J. Newman. Modularity and community structure in networks. *PNAS*, 103(23):8577–8582, 2006.

[83] Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, 76(3):036106, 2007.

[84] Laurens J. P. van der Maaten and Geoffrey E. Hinton. Visualizing data using t-sne. *JMLR*, 9: 2579–2605, 2008.

[85] Roderick P McDonald. *Test theory: A unified treatment*. Psychology Press, 2013.

[86] Walter M. Bonivento. The SHiP experiment at CERN. *Journal of Physics: Conference Series*, 878:012014, jul 2017. doi: 10.1088/1742-6596/878/1/012014.

[87] Cliff Burgess and Guy Moore. *The Standard Model: A Primer*. Cambridge University Press, 2006. doi: 10.1017/CBO9780511819698.

[88] A Baranov, E Burnaev, D Derkach, A Filatov, N Klyuchnikov, O Lantwin, F Ratnikov, A Ustyuzhanin, and A Zaitsev. Optimising the active muon shield for the ship experiment at cern. In *Journal of Physics: Conference Series*, volume 934, page 012050. IOP Publishing, 2017.

[89] Jonas Mockus. Application of bayesian approach to numerical methods of global and stochastic optimization. *Journal of Global Optimization*, 4(4):347–365, 1994.

[90] Jonas Mockus, Vytautas Tiesis, and Antanas Zilinskas. Toward global optimization, volume 2, chapter bayesian methods for seeking the extremum, 1978.

[91] Emmanuel Vazquez and Julien Bect. Convergence properties of the expected improvement algorithm with fixed mean and covariance functions. *Journal of Statistical Planning and inference*, 140(11):3088–3095, 2010.

[92] Tom Inglis. *Directional drilling*, volume 2. Springer Science & Business Media, 2013.

[93] H. Zhou, P. Hatherly, S. Monteiro, F. Ramos, F. Oppolzer, and E. Nettleton. A hybrid gp regression and clustering approach for characterizing rock properties from drilling data. *Technical Report ACFR-TR-2011-001*, 2010.

[94] H. Zhou, P. Hatherly, F. Ramos, and E. Nettleton. An adaptive data driven model for characterizing rock properties from drilling data. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1909–1915. IEEE, 2011.

[95] C. Hegde and K.E. Gray. Use of machine learning and data analytics to increase drilling efficiency for nearby wells. *Journal of Natural Gas Science and Engineering*, 40:327–335, 2017.

[96] D. LaBelle, J. Bares, and I. Nourbakhsh. Material classification by drilling. In *Proceedings of the International Symposium on Robotics and Automation in Construction, Taipei, Taiwan*, 2000.

[97] D. LaBelle. *Lithological classification by drilling.* Carnegie Mellon University, The Robotics Institute, 2001.

[98] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference and prediction.* Springer, 2 edition, 2009.

[99] Nikita Klyuchnikov, Alexey Zaytsev, Arseniy Gruzdev, Georgiy Ovchinnikov, Ksenia Antipova, Leyla Ismailova, Ekaterina Muravleva, Evgeny Burnaev, Artyom Semenikhin, Alexey Cherepanov, et al. Data-driven model for the identification of the rock type at a drilling bit. *Journal of Petroleum Science and Engineering*, 178:506–516, 2019.

[100] Arthur Zimek and Erich Schubert. Outlier detection. *Encyclopedia of Database Systems*, pages 1–5, 2017.

[101] T Diomede, F Nerozzi, T Paccagnella, and E Todini. The use of meteorological analogues to account for lam qpf uncertainty. *Hydrology and Earth System Sciences*, 12(1):141–157, 2008.

[102] P.J. Moore and M.A. Little. Enhancements to a method of analogues forecasting algorithm. *NOLTA2014*, pages 317–320, 2014.

[103] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, 41(3):1–58, 2009.

[104] Luís MA Bettencourt, Ruy M Ribeiro, Gerardo Chowell, Timothy Lant, and Carlos Castillo-Chavez. Towards real time epidemiology: data assimilation, modeling and anomaly detection of health surveillance data streams. In *NSF Workshop on Intelligence and Security Informatics*, pages 79–90. Springer, 2007.

[105] SP King, DM King, K Astley, L Tarassenko, P Hayton, and S Utete. The use of novelty detection techniques for monitoring high-integrity plant. In *Proceedings of the International Conference on Control Applications*, volume 1, pages 221–226. IEEE, 2002.

[106] Ayesha Arjumand Nayeem, Ramachandran Venkatesan, and Faisal Khan. Monitoring of downhole parameters for early kick detection. *Journal of Loss Prevention in the Process Industries*, 40: 43–54, 2016.

[107] Evgeniya Romanenkova, Alexey Zaytsev, Nikita Klyuchnikov, Arseniy Gruzdev, Ksenia Antipova, Leyla Ismailova, Evgeny Burnaev, Artyom Semenikhin, Vitaliy Koryabkin, Igor Simon, et al. Real-time data-driven detection of the rock-type alteration during a directional drilling. *IEEE Geoscience and Remote Sensing Letters*, 17(11):1861–1865, 2019.

[108] Mahmoud Reza Saybani, Teh Ying Wah, Amineh Amini, and Saeed Reza Aghabozorgi Sahaf Yazdi. Anomaly detection and prediction of sensors faults in a refinery using data mining techniques and fuzzy logic. *Scientific Research and Essays*, 6(27):5685–5695, 2011.

[109] Mark van der Meijde, HMA Van Der Werff, PF Jansma, Freek D van der Meer, and GJ Groothuis. A spectral-geophysical approach for detecting pipeline leakage. *International Journal of Applied Earth Observation and Geoinformation*, 11(1):77–82, 2009.

[110] Maria Kontaki, Apostolos N Papadopoulos, and Yannis Manolopoulos. Similarity search in time series databases. In *Encyclopedia of Database Technologies and Applications*, pages 646–651. IGI Global, 2005.

[111] Joan Serra and Josep Ll Arcos. An empirical evaluation of similarity measures for time series classification. *Knowledge-Based Systems*, 67:305–314, 2014.

[112] Anthony Bagnall and Jason Lines. An experimental evaluation of nearest neighbour time series classification. *arXiv preprint arXiv:1406.4757*, 2014.

[113] Tapas Kanungo, David M Mount, Nathan S Netanyahu, Christine D Piatko, Ruth Silverman, and Angela Y Wu. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 1(7):881–892, 2002.

[114] Yizong Cheng. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):790–799, 1995.

[115] William HE Day and Herbert Edelsbrunner. Efficient algorithms for agglomerative hierarchical clustering methods. *Journal of Classification*, 1(1):7–24, 1984.

[116] Animesh Patcha and Jung-Min Park. An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer Networks*, 51(12):3448–3470, 2007.

[117] Salima Omar, Asri Ngadi, and Hamid H Jebur. Machine learning techniques for anomaly detection: an overview. *International Journal of Computer Applications*, 79(2):33–41, 2013.

[118] Martin T Hagan, Howard B Demuth, Mark H Beale, and Orlando De Jesús. *Neural network design*, volume 20. Pws Pub. Boston, 1996.

[119] Anup K Ghosh, Christoph Michael, and Michael Schatz. A real-time intrusion detection system based on learning program behavior. In *International Workshop on Recent Advances in Intrusion Detection*, pages 93–109. Springer, 2000.

[120] James Cannady. Artificial neural networks for misuse detection. In *National Information Systems Security Conference*, volume 26, pages 443–456. Baltimore, 1998.

[121] Weizhong Yan and Lijie Yu. On accurate and reliable anomaly detection for gas turbine combustors: A deep learning approach. In *Proceedings of the annual Conference of the Prognostics and Health Management Society*, pages 1–8, 2015.

[122] Steven A Hofmeyr, Stephanie Forrest, and Anil Somayaji. Intrusion detection using sequences of system calls. *Journal of Computer Security*, 6(3):151–180, 1998.

[123] Yu V Vadetskii. Drilling oil and gas wells. *Nedra, Moscow*, 1983.

[124] Roman J Shor, Mitch Pryor, and Eric Van Oort. Drillstring vibration observation, modeling and prevention in the oil and gas industry. In *ASME 2014 Dynamic Systems and Control Conference*, pages V003T37A004–V003T37A004. American Society of Mechanical Engineers, 2014.

[125] Robert D Grace. *Blowout and well control handbook*. Gulf Professional Publishing, 2017.

[126] Evgeny Burnaev, Pavel Erofeev, and Artem Papanov. Influence of resampling on accuracy of imbalanced classification. In *Eighth International Conference on Machine Vision (ICMV 2015)*, volume 9875, page 987521. International Society for Optics and Photonics, 2015.

[127] Nataliia Kozlovskaia and Alexey Zaytsev. Deep ensembles for imbalanced classification. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 908–913. IEEE, 2017.

[128] Oded Maimon and Lior Rokach. *Data mining and knowledge discovery handbook*. Springer, 2005.

[129] Bradley Efron and Robert Tibshirani. Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. *Statistical Science*, pages 54–75, 1986.

[130] Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren, editors. *Automated Machine Learning: Methods, Systems, Challenges*. Springer, 2018. In press, available at http://automl.org/book.

[131] Steve Branson, Grant Van Horn, and Pietro Perona. Lean crowdsourcing: Combining humans and machines in an online system. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7474–7483, 2017.

[132] Christine Rzepka. Examining the use of voice assistants: A value-focused thinking approach. In *25th Americas Conference on Information Systems, AMCIS 2019, Cancún, Mexico, August 15-17, 2019*. Association for Information Systems, 2019.

[133] Bochao Wang, Hang Xu, Jiajin Zhang, Chen Chen, Xiaozhi Fang, Ning Kang, Lanqing Hong, W. Zhang, Yong Li, Zhicheng Liu, Zhenguo Li, Wenzhi Liu, and Tong Zhang. Vega: Towards an end-to-end configurable automl pipeline. *ArXiv*, abs/2011.01507, 2020.

Skoltech

Skolkovo Institute of Science and Technology

2021