NATIONAL RESEARCH UNIVERSITY
HIGHER SCHOOL OF ECONOMICS

as a manuscript

Kirill Struminsky

LEARNING GAURANTEES AND EFFICIENT INFERENCE FOR STRUCTURED PREDICTION

PHD DISSERTATION SUMMARY
for the purpose of obtaining academic degree
Doctor of Philosophy in Computer Science

Moscow — 2023

The PhD Dissertation was prepared at National Research University Higher School of Economics.

Academic Supervisor: Dmitry P. Vetrov, Candidate of Science, National Research University Higher School of Economics.

1 Introduction

Machine learning attempts to recover and describe empirical relationships in data. Often the interest is in quantifying or attributing observed data to a predetermined set of categories. For example, how does the price of an apartment depend on its location and parameters? Will the user want to read this email? These questions can be answered based on historical data containing details of past transactions or the history of user interaction with previously received emails. Attribution can also be of interest when the attributes are not known in advance: is it possible, for example, to distinguish several distinctive categories in the data?

At the same time, in applications there are problems in which the desired dependencies fall outside the scope of the examples described above. If, for example, we are talking about a machine translation task, then each text in the source language must be matched with a text in the target language. In this case, it would be incorrect to represent the predicted translation as a number or an element of the set of all possible translations. On the contrary, it would be convenient to represent the text as a sequence of words, where the translation algorithm must predict each word, focusing both on the original sentence and on neighboring words of the translation.

Variables in the data, represented as a set of mutually dependent values, are usually called structured. The area of machine learning devoted to the prediction of structured variables is called structured prediction. A characteristic feature of structured variables is the combinatorial growth of the number of possible values (outcomes) depending on the parameters of the problem. Ignoring the nuances of the problem, in the machine translation example, with a dictionary size of w and a known translation length l, the algorithm must choose among w^l possible translations. This feature raises questions about learning guarantees and efficient inference. Namely, how many examples are enough to reliably restore the required dependence? How to quickly select an element from a possible set of outcomes? This work is devoted to the study of these issues.

1.1 Relevance

As the area of machine learning application keeps spreading[56], the variety of tasks and problem setups is also growing, making structured prediction more in demand. In particular, the deep learning developments have made it possible to bring algorithms for natural language processing and computer vision to a qualitatively new level. In supervised learning problems where the target variables are structured variables, learning is often reduced to minimizing the cross-entropy loss function. Such a loss function, in turns, requires defining the distribution of a target structured variable. For example, in natural language processing tasks, distribution over text outputs is introduced by factorizing the distribution into word-level distributions according to the chain rule (for more details, see [23, Chapter 10]). Another solution, common, for example, in the problem of semantic segmentation, is to assume that all elements of a structured variable are independent given the input image (as, for example, done in [54]). Recent studies are mostly devoted to the design of neural network architectures for parameterizing distributions in the described approach, as well as scaling the described approach [8, 30, 72]. Structured outputs prompted such key developments as recurrent [28, 62] and convolutional neural networks [22, 43], as well as transformers [67] for sequence processing, UNet architecture for image processing [54].

The disadvantage of the above approach to structured prediction and deep learning

in general is the limited interpretability of the recovered dependencies. In the meantime, certain governmental regulators introduce the "right to explanation"[68], according to which a person can demand an explanation of how the machine learning system made a decision regarding him. Thus, the problem of interpreting machine learning algorithms becomes especially acute with the development of deep learning systems. As a result, a designated area of research has emerged, attempting to interpret specific architectures [69, 31, 52], as well as to develop interpretation recipes for arbitrary machine learning algorithms [37, 51, 11]. At the same time, the idea of using latent structured variables to increase the interpretability of machine learning algorithms has gained popularity [32, 39]. Next, we describe the idea in more detail. Deep neural networks comprise a sequence of elementary computing blocks, however the combined output of these blocks is difficult to interpret. On the other hand, network evaluation may be more transparent if some of these intermediate construction blocks have interpretable (structured) outputs, and the network architecture itself takes into account the problem specifics. For example, in a sentiment analysis task one can design a model that chooses a small subset of words, based on which the model will make a prediction. In practice, the words chosen by such a model help to interpret the output. Besides that, neural networks with latent structured variables can be seen as an evolution of latent variable models such as hidden Markov chains [12] or probabilistic context-free grammars [55] for modeling languages by adding more expressive neural network models.

However, in the case of discrete latent variables, the standard training approaches based on backpropagation is not applicable due to the non-differentiability of the block that returns the latent variable. The solution to this problem usually comes down to heuristic gradient substitutes [4] or stochastic relaxation [29, 38, 5, 45]. One of the chapters of this work is devoted to the problem of learning with hidden permutations. Another problem related to latent structured variables, which does not lose its relevance to this day, is the design of architectures with latent variables and the choice of objective functions. As previous work indicates [33, 16], end-to-end learning in such models often leads to predictive models that ignore hidden variables, learning the dependence only on the basis of standard neural network components. The standard solution in this case is learning with partial labeling of latent variables: for a subset of training samples, an additional loss function is introduced to encourage the desired prediction. An alternative would be to choose an architecture that does not allow for sufficient prediction accuracy without using the hidden variable [11].

Along with the development of practical approaches and algorithms for working with structural variables, it is important to obtain guarantees on the quality of their work. In the context of structured prediction, the combinatorial growth in the number of possible predictions and the unequal contribution of erroneous predictions (not all inaccurate predictions are equally bad) are the two factors that distinguish structured prediction from the well-studied classification setup [44]. Generalization in the context of structured prediction is discussed in [17, 36]. In practice, target metric often does not coincide with the functional being optimized during training (a surrogate loss function); a number of results on relationship between target and surrogate losses have been obtained for structural prediction problems. In the paper [14], the authors showed the consistency of a class of quadratic surrogate loss functions, and the paper [44] obtained an estimate for the discrepancy between the accuracy of the prediction according to the target metric and the surrogate loss function. Later, [42] generalized these results to smooth convex surrogate loss functions. The above works assume that the surrogate loss function is consistent, although inconsistent surrogates are also often used in practice: for example, the multi-class

support vector machine in the Crammer-Singer form [19], as well as its generalizations to structured variables [63, 65]. As part of the study of inconsistent loss functions, this dissertation generalized the results [44] by obtaining estimates for quadratic surrogate loss functions without the additional requirement of consistency.

1.2 Work Goals

As noted above, structured variables often arise in various machine learning applications. Prospective problem setups may include structured target variables in the case of supervised learning, as well as structured latent variables in both supervised and unsupervised setups. In addition to prediction quality metrics, inference speed becomes a critical performance aspect as we shift to structured variables with a combinatorial number of possible outcomes. The goal of this work was to develop structured prediction methods that meet the requirements arising in applications: to develop structured prediction methods for observed and latent structured variables, while emphasising algorithms with feasible inference time and the availability of learning guarantees for the proposed methods.

Within the framework of the goals described above, the following **tasks** were set:

1. development prediction methods for such structural variables as permutations and subsets of a given size,
2. study of consistency and derivation of learning guarantees for supervised learning tasks with a structured target variable,
3. development and empirical analysis of models with latent structural variables,
4. development of efficient inference methods for structured latent variables
5. the use of latent structured variables for data interpretation, as well as the construction of interpretable machine learning methods.

Contributions. When solving the tasks above, we obtained the following results.

1. We developed and evaluated a gradient-based method to optimize over a set of permutations or subsets.
2. In supervised structured prediction setup, we carried out analysis of quadratic surrogate loss functions and quantified surrogate consistency in a novel setting.
3. We proposed and studied several approaches to recovering latent structured variables based on maximum evidence principle and quadratic surrogate loss functions.
4. We proposed a number of efficient inference procedures for such latent structured variables as permutations and fixed-size subsets.
5. We developed methods for interpreting data based on latent structured variables.

1.3 Practical Applications

The developed approach to permutation optimization is applicable for restoring the structure of the relationship between variables in data, which, in particular, is in demand when interpreting machine learning models. The prior distribution for convolutional neural network parameters offers a method for rapidly adapting model parameters to a new adjacent data domain. The method for estimating the parameters of a multi-user communication

channel finds application in modern cellular networks. A probabilistic model for preprocessing geophysical exploration data provides a convenient way to detect anomalies and recover gaps in historical data.

1.4 Methodology

Our theoretical analysis of structured prediction is based on sections of probability theory, statistical learning theory, and optimization. In a general structured prediction setup, we obtained a result applicable to a number of structured prediction problems. Other consideration are based on probabilistic machine learning formalism, as well as the Bayesian approach to machine learning. The proposed methods are based on the basic sections of probability theory and stochastic optimization. Besides a few rigorous proofs, this work mostly relies on the empirical evaluation methods. We implemented the proposed algorithms in Python, assessed their performance and compared with analogues on synthetic and real data sets.

1.5 Publications and Probation of the Work

First-tier publications:

1. **Struminsky K.**, Lacoste-Julien S., Osokin A. Quantifying Learning Guarantees for Convex but Inconsistent Surrogates //Advances in Neural Information Processing Systems. – 2018. – С. 669-677. *Contribution of the thesis author:* A general lower bound on the calibration function in structured prediction setup; calculation of the lower bound coefficients for hierarchical classification; calculation of the lower bound coefficients for ranking.

2. Gadetsky, A., **Struminsky, K.**, Robinson, C., Quadrianto, N., & Vetrov, D. P. (2020). Low-Variance Black-Box Gradient Estimates for the Plackett-Luce Distribution. In AAAI (pp. 10126-10135). *Contribution of the thesis author:* An approach to optimization over permutations and acyclic graphs based on variational optimization for Plackett-Luce distributions; generalization of the RELAX gradient estimator to the case of the Plackett-Luce distribution.

3. Atanov, A., Ashukha, A., **Struminsky, K.**, Vetrov, D., & Welling, M. (2018, September). The Deep Weight Prior. In International Conference on Learning Representations. *Contribution of the thesis author:* Adaptation of the variational auto-encoder to the problem of estimating the prior distribution on the parameters of the Bayesian neural network.

Standard-tier publications:

1. **Struminsky K.** et al. A new approach for sparse Bayesian channel estimation in SCMA uplink systems //2016 8th International Conference on Wireless Communications & Signal Processing (WCSP). – IEEE, 2016. – С. 1-5. *Contribution of the thesis author:* Probabilistic model for estimating the parameters of a multi-user communication channel; improved scheme for approximate inference of parameters of a multi-user communication channel and estimation of the channel configuration.

2. **Struminskiy K.** et al. Well Log Data Standardization, Imputation and Anomaly Detection Using Hidden Markov Models //Petroleum Geostatistics 2019. – European Association of Geoscientists & Engineers, 2019. – Т. 2019. – №. 1. – С. 1-5.

Contribution of the thesis author: A probabilistic model for the preprocessing of geological and physical exploration data.

In all papers, with the exception of "The Deep Weight Prior" [1], the applicant is the main author.

Conference presentations and seminar talks:

1. Bayesian Deep Learning Workshop, NeurIPS 2019, Vancouver, Canada, 13 December, 2019.

   Topic: Low-variance Gradient Estimates for the Plackett-Luce Distribution (spotlight presentation, poster).

2. 8th International Conference on Wireless Communications and Signal Processing, Yangzhou, Chine, 13-15 Ocboter, 2016.

   Topic: A new approach for sparse Bayesian channel estimation in SCMA uplink systems (oral presentation).

3. Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20), New York, USA, 7-12 February, 2020.

   Topic: Low-Variance Black-Box Gradient Estimates for the Plackett-Luce Distribution (oral presentation, poster).

4. EAGE Conference on Petroleum Geostatistics, Florence, Italy, 2-6 September, 2019.

   Topic: Well Log Data Standardization, Imputation and Anomaly Detection Using Hidden Markov Models (oral presentation).

5. Thirty-second Annual Conference on Neural Information Processing Systems (NeurIPS 2018), Montral, Canada, 2-8 December, 2018.

   Topic: Quantifying Learning Guarantees for Convex but Inconsistent Surrogates (poster).

6. Thirty-fifth Annual Conference on Neural Information Processing Systems (NeurIPS 2021), online, 6-14 December, 2021.

   Topic: Leveraging Recursive Gumbel-Max Trick for Approximate Inference in Combinatorial Spaces (poster).

7. Seventh International Conference on Learning Representations (ICLR 2019), New Orlean, USA, 6-9 May, 2019.

   Topic: The Deep Weight Prior (poster).

8. Bayes Group Research Seminar, Moscow, Russia, 26 October, 2018.

   Topic: Quantifying Learning Guarantees for Convex but Inconsistent Surrogates (oral presentation).

9. Sberbank Data Science Journey, Moscow, Russia, 10 November, 2018.

   Topic: Quantifying Learning Guarantees for Convex but Inconsistent Surrogates (oral presentation, poster).

10. Machines Can See: Computer Vision and Deep Learning Summit, Moscow, Russia, 25 June, 2019.

    Topic: The Deep Weight Prior (poster).

11. International Conference on Analysis of Images, Social Networks and Texts, AIST 2019, Kazan, Russia, 17-19 Jule, 2019.

    Topic: A Simple Method to Evaluate Support Size and Non-uniformity of a Decoder-Based Generative Model (oral presentation).

12. Advances in Approximate Bayesian Inference, NIPS 2016 Workshop, Barcelona, Spain, 2016.

    Topic: Robust Variational Inference (poster).

2 Preliminaries

2.1 Structured Variables in Machine Learning

We start by introducing the concept of a structured variable. In machine learning, a structured variable is an umbrella term for random variables, united by the following characteristic properties. Firstly, a structured variable has a large number of possible values: the support of a random variable is typically a finite set that cannot be quickly enumerated on a computer. Secondly, these variables are presented as a set of mutually dependent random variables. The second property can act as a definition of a structured variable. For clarity, we turn to specific examples below.

In applications, structured variables can act as a target variable in supervised problems (structured prediction), and can also act as an auxiliary latent variable in models with latent variables.

One of the standard examples of a structured prediction problem is segmentation in computer vision [34]. In this case, the structured variable is the segmentation mask of an image. Segmentation mask components are mutually dependent, since close points of an image with high probability correspond to the same class. Other examples of structured prediction problems include ranking [10, 49], extreme classification [13]. Many natural language processing tasks are also structured prediction tasks. A model that produces a text output, whether it is a summation, a translation, or an answer to a question, must predict a sequence of interdependent random variables. In deep learning, such models are defined by the seq2seq architecture [62], and for prediction they use approximate search algorithms among all possible options [50].

Before the spread of deep learning methods, structured variables were also in demand in natural language processing tasks, often playing the role of auxiliary latent variables there [58]. For example, early machine translation algorithms could rely on the input sentence parse tree to better convey the sentence meaning. In this example, the structured variable is the sentence parse tree, and a separate auxiliary model trained on different data could be used to build the tree.

However, these days machine learning solutions rarely rely on pipelines built with auxiliary models and tasks. Instead, deep neural networks allow end-to-end learning, pre-training on unlabeled data [41, 20], and knowledge transfer to small datasets [74]. As a result, end-to-end learning in models with latent structured variables became a relevant research topic. Such models allow to take the best of both worlds: on the one hand, the flexibility of neural networks, on the other hand, reliance on prior knowledge through structured variables for better interpretability and more efficient use of data.

Popular models with latent structured variables include hidden Markov chains [48] with sequence markup as a structured variable, probabilistic context-free grammars [15] with a parse tree as a structured variable, and a temporal sequence classification model [25] with

latent sequence segmentation mask. These models are based on limiting assumptions on the model variables that are necessary for efficient learning an inference. More recent approaches circumvent the limiting assumptions by relying on stochastic gradient descent for end-to-end learning and fast amortized inference [45]. Some of the examples include models with hidden parse trees [16], implicit feature subset selection [11], and hidden text generation order [27].

In the next section, we introduce a general supervised structured prediction setup.

2.2 Structured Prediction Basics

Let us first consider the standard structured prediction setup. Namely, consider a supervised learning problem with inputs $x \in \mathcal{X}$ from an arbitrary set $\mathcal{X}$, and the goal is to predict a structured variable $y \in \mathcal{Y}$, which takes values in a finite set $\mathcal{Y}$. The data is distributed according to law $\mathcal{D}$, and y is a realization of a random vector Y with support $\mathcal{Y} \subset \mathbb{R}^m$. In general, the label of a training sample lie in a $\hat{\mathcal{Y}}$, which can differ from $\mathcal{Y}$.

To define a prediction algorithm, we define a function $f : \mathcal{X} \to \mathbb{R}^{|\mathcal{Y}|}$ that assigns a score to each possible structure $y \in \mathcal{Y}$ and then chooses the optimal structure as a prediction

$$\text{pred}(f(x)) := \arg\max_{y \in Y} f_y(x). \tag{1}$$

The difference between structured prediction setup and supervised learning setup is that the set of possible outcomes $\mathcal{Y}$ is large due to the combinatorial growth of possible outcomes. For example, in ranking, the outcome can be a permutation of elements, and when segmenting, a sequence of class labels. Therefore, the model should offer a quick way to solve the problem 1 at inference stage. In addition, we need to store function f in memory. Typically, one resorts to a low-rank parameterization of the function $f(x) = Fg(x)$, where $F : \mathbb{R}^d \to \mathbb{R}^{|\mathcal{Y}|}$ is a fixed linear operator and $g(x) : \mathcal{X} \to \mathbb{R}^d$ is a function with we construct during training. Such a parameterization allows to reduce the memory footprint as we only have to store a function with $d \ll \|\mathcal{Y}\|$ outputs and allows to design efficient algorithms for inference task 1 that rely on the choice of matrix F. At the same time, the parameterization limits the set of possible predictions, since the score vector $f(x)$ lies within the linear span of the columns of the matrix $\mathcal{F} = \operatorname{span} F$. Below we refer to $\mathcal{F}$ as the set of feasible scores.

Given a loss function $L(\cdot, \cdot) : \mathcal{Y} \times \hat{\mathcal{Y}} \to \mathbb{R}$ quantifying prediction quality, the goal is to find f that is optimal in terms of risk (the expected value of the loss):

$$\mathcal{R}_L(f) := \mathbb{E}_{X,Y} L(\text{pred}(f(X)), Y). \tag{2}$$

Direct optimization of risk is often unfeasible (in particular, a finite sample approximation of $R_L(f)$ is not differentiable with respect to f outputs). For optimization, we introduce an auxiliary (surrogate) loss function $\Phi : \mathbb{R}^k \times \mathcal{Y} \to \mathbb{R}$ and define the surrogate risk

$$\mathcal{R}_\Phi(f) := \mathbb{E}_{X,Y} \Phi(f(x), y). \tag{3}$$

We emphasize that the function 3 takes the value of $f(x)$ as an argument and makes gradient optimization feasible, whereas the objective function 2 takes the predictions from a discrete set. Popular surrogate loss functions include quadratic functions [14, 7], likelihood-based functions [34] and the surrogates arising in variations of SVM [53, 65].

When replacing the objective function with a surrogate one, the question inevitably arises of the relationship between the optimum of the surrogate loss function and the solution of the original problem. To answer this question, the concept of consistency of a

surrogate loss function was introduced [2]. The concept is closely related to the concept of Fisher consistency [18, p.287]. Intuitively, a surrogate loss function is consistent if the optimal f w.r.t. the surrogate risk is also optimal with respect to the original risk.

Let us define consistency in terms of the calibration function that connects the surrogate and target loss functions. For a score $f \in \mathcal{F} \subseteq \mathbb{R}^k$ and a distribution $q \in \Delta_k$ on a set of possible outcomes $Y \sim q$, we introduce the conditional risk $l(f, q) := \mathbb{E}_Y L(\text{pred}(f), Y)$ and the conditional surrogate risk $\phi(f, q) := \mathbb{E}_Y \Phi(f, Y)$. The excess (surrogate) risk is the deviation of $\delta l(f, q)$ ($\delta\phi(f, q)$) from the optimal risk

$$\delta l(f, q) := l(f, q) - \inf_{\hat{f} \in \mathcal{F}} l(\hat{f}, q) \tag{4}$$

$$\delta\phi(f, q) := \phi(f, q) - \inf_{\hat{f} \in \mathcal{F}} \phi(\hat{f}, q). \tag{5}$$

Using these auxiliary functions, we define the calibration function.

Definition 1. For a loss function L, a surrogate loss function Φ, and a set of feasible scores $\mathcal{F}$, the calibration function $H_{\Phi,L,\mathcal{F}}(\varepsilon)$ at the argument $\varepsilon \geq 0$ is equal to the surrogate risk infimum given the target risk is not less than ε:

$$H_{\Phi,L,\mathcal{F}}(\varepsilon) := \inf_{f \in \mathcal{F}, q \in \Delta_k} \delta\phi(f, q) \tag{6}$$

$$\text{s.t.} \delta l(f, q) \geq \varepsilon \tag{7}$$

Intuitively, the calibration function estimates how small the error of the surrogate loss function can be for a fixed loss function value. When loss is high due to prediction error, a consistent surrogate loss function should also return a high value. The following theorem connects the surrogate risk and the target risk using the calibration function.

Theorem 1 (Associating $\mathcal{R}_L$ with $\mathcal{R}_\Phi$ through the calibration function)**.** *Let $H_{\Phi,L,\mathcal{F}}$ be the calibration function for the loss function L and the surrogate loss function Φ, and let $\mathcal{F}$ be the set of feasible scores. Let $\Phi, \hat{L}, \mathcal{F}$ be a convex non-decreasing lower bound for the calibration function $H_{\Phi,L,\mathcal{F}}$. Assume additionally that Φ is a continuous function bounded from below. Then for any $\varepsilon > 0$ such that $\hat{H}_{\Phi,L,\mathcal{F}}$ is finite and any score $f \in \mathcal{F}$ holds*

$$\mathcal{R}_\Phi(f) < \inf_{\hat{f} \in \mathcal{F}} R_\phi(\hat{f}) + \hat{H}_{\Phi,L,\mathcal{F}}(\varepsilon) \Rightarrow \mathcal{R}_L(f) < \inf_{\hat{f} \in \mathcal{F}} \mathcal{R}_L(\hat{f}) + \varepsilon. \tag{8}$$

Next, we define η-consistency of a surrogate loss function, which is inspired by the above theorem.

Definition 2 (η-consistency of a surrogate loss function)**.** A surrogate loss function Φ is consistent up to the level $\eta \geq 0$ (η-consistent) for the objective function L and the set of feasible scores $\mathcal{F}$ if and only if the calibration function satisfies $H_{\Phi,L,\mathcal{F}}(\varepsilon) > 0$ for any $\varepsilon > \eta$ and there exists $\hat{\varepsilon} > \eta$ such that that $H_{\Phi,L,\mathcal{F}}(\hat{\varepsilon})$ is finite.

For $\eta = 0$, the above definition coincides with the notion of consistency common in the machine learning literature [35, 47]. Thus, one can validate consistency of a surrogate loss by showing that the calibration function is positive in the punctured neighborhood of $\varepsilon = 0$. However, in practice consistency may not be sufficient to build realistic learning guarantees. As Osoking et al. [44] showed, for various problems in structured prediction the theorem 1 delivers non-trivial guarantees only for practically unattainable surrogate

risk values. It turns out that the scale of the calibration function plays an important role as well.

The notion of η-consistency is crucial for the results presented in this work. First, it allows to obtain learning guarantees in theorem 1 under a weaker assumption of inconsistent surrogate functions, that is, not falling under the definition of 0-consistency. Second, we construct a tighter calibration function lower bound for inconsistent surrogate losses and obtain a more optimistic learning guarantees.

2.3 Probabilistic Approach to Structured Prediction

The structured prediction setup in Section 2.2 used the language of probability to introduce assumptions about the data and reformulate learning as an optimization task. In addition, the language of probability is a convenient tool for defining non-deterministic prediction models and for modeling various modes of uncertainty such as uncertainty in the choice of model and uncertainty in the prediction of a particular model. *Probabilistic machine learning* is an approach to machine learning that relies on probability theory to formulate and solve machine learning tasks. Next, we describe this approach in more detail, starting from common examples of its application.

When formulating a problem within the framework of the probabilistic approach, the first step is to choose a set of random variables appearing in the problem, as well as their joint distribution. Treating the data as random variables, we describe the desired patterns by choosing the appropriate class of distributions.

So, for example, when building a logistic regression model, the class label $y \in \mathcal{Y} = \{-1, 1\}$ is represented as a random variable Y with a Bernoulli distribution depending on the input object $x \in \mathcal{X} = \mathbb{R}^d$, and probability $\mathbb{P}_Y(y \mid x; \theta) = \frac{1}{1+\exp(y\theta^T x)}$ with parameters $\theta \in \mathbb{R}^d$. Following the assumptions about the data, we assume that tuples of objects x and labels y are jointly independent (i.e., data is i.i.d.). Label distribution allows to model the uncertainty in the prediction of the model, which may be due to lack of data, the inflexibility of the model or the label being non-deterministic. If uncertainty also arises when estimating the model parameters, the probabilistic approach allows us to consider the model parameters as a random variable Θ as well. In the absence of any knowledge about the values of the Θ parameter, its distribution can be assumed to be normal $\Theta \sim \mathcal{N}(\mid 0, \operatorname{diag}\sigma), \sigma \in \mathbb{R}^d$ with a diagonal covariance matrix with parameters $\sigma \in \mathbb{R}^d$. Assuming $\Theta \perp\!\!\!\perp Y$, we get the joint distribution of Θ parameters and Y labels. The interpretation of model parameters as random variables underlies the Bayesian approach, allowing one to estimate the uncertainty in choosing model parameters using the posterior distribution $p_\Theta(\theta \mid (x^i, y^i)_{i=1}^n; \sigma)$ for a data set of n objects.

Note that the above example did not make any assumptions about the distribution of the input x, as they are not necessary when considering the classification problem. Such models are called discriminative. At the same time, the probabilistic approach makes it possible to oppose discriminate models to generative ones, which also model the distribution of input objects. A classic example of a generative model is the naive Bayes classifier. It is based on the joint distribution $p(x, y \mid \theta) := p(x \mid y; \theta)p(y; \theta)$, and when classifying objects it relies on the conditional distribution $p(y \mid x, \theta)$.

In addition, the probabilistic approach allows you to introduce additional random variables, making it possible to simplify the description of the desired dependencies. For example, Latent Dirichlet Allocation[6] model for texts groups objects $x \in \mathcal{X}$ according to topics: for a text corpus, the model defines a set of $\tau \in \mathbb{N}$ topics, and then represents each individual text $p(x \mid t)$ based on a vector of topics $t \in \Delta^T$ that are reflected in the

text. An auxiliary random variable in this case is a set of topics in the text t with a priori distribution $p_T(t)$. Since such auxiliary quantities are not reflected in the data, they are commonly referred to as *latent variables*.

The choice of a joint distribution often leads to the choice of a training method. Among the possible training methods, we distinguish two categories: in the case when the choice of model parameters is of interest, the parameters can be obtained by maximizing the likelihood:

$$\max_{\theta} \log p(\{x^i, y^i\}_{i=1}^n \mid \theta) \tag{9}$$

If there are latent variables in the model, it is natural to consider the marginal likelihood instead. In the literature, this approach is referred to as empirical Bayes or type-II maximum likelihood:

$$\max_{\theta} \log p(\{x^i, y^i\}_{i=1}^n \mid \theta) \tag{10}$$

$$\log p(\{x^i, y^i\}_{i=1}^n \mid \theta) = \log \mathbb{E}_T p(\{x^i, y^i\}_{i=1}^n, T \mid \theta). \tag{11}$$

In the case when one of the hidden quantities is of interest, one can restore their characteristic values based on the posterior distribution $p(T \mid \{x^i, y^i\}_{i=1}^n, \theta)$. In particular, the posterior distribution can be used to solve the problem 10. It is often impossible to calculate the posterior distribution explicitly in practice, and one of the common approaches to its approximation is the variational inference, which reduces the task to the optimization problem

$$\max_{\phi} \mathbb{E}_T \log \frac{p(\{x^i, y^i\}_{i=1}^n, T \mid \theta)}{q(T \mid \ phi)}, \tag{12}$$

where the expectation is taken over the random variable T with the distribution $q(\cdot \mid \phi)$, and the optimization is performed over the distribution parameters ϕ.

The objective functions described above can be interpreted as surrogate loss functions introduced using the probabilistic approach. Since surrogate functions eqs. (9), (10) and (12) do not depend on the loss function $L(\cdot, \cdot)$ in any way, these objectives may be inconsistent. Besides that, in comparison with the classical formulation of structured learning, the probabilistic approach allows to operate with latent structured variables. This, in turn, allows to design and train in an "end-to-end" fashion prediction models that involve auxiliary structured latent variables as intermediate components. For example, when solving a discriminative task for texts, the parse trees of sentences can be incorporated as a hidden auxiliary variable that provides additional information for the final prediction.

There are a number of general methods for solving the problems described above. In particular cases, there exists an analytical solution for problems eqs. (9), (10) and (12). In the general case, when an analytical solution is not available, approximate solution can be found using stochastic optimization methods. Problem 9 can be reduced to stochastic optimization in the case when the parameter θ ranges over a discrete structured set and we are unable to iterate through the whole domain containing θ. In problems eqs. (10) and (12), stochastic optimization allows you to optimize the mathematical expectation in the problem statement without resorting to its exact calculation. The two main approaches to constructing unbiased gradient estimates are the reparameterization trick and the REINFORCE algorithm. For structured variables, the first is rarely applicable, and the second often requires careful tuning for each problem.

Below we describe the two main approaches to estimating stochastic gradients. For

the problem of a form

$$\max_{\theta} \mathbb{E}_T f(T), \tag{13}$$

where the random variable T has the distribution $q(\cdot \mid \theta)$, the REINFORCE algorithm constructs an unbiased gradient estimator by using the log-derivative trick

$$\nabla_\theta \mathbb{E}_T f(T) = \mathbb{E}_T f(T) \nabla_\theta \log q(T \mid \theta), \tag{14}$$

which allows us to construct an unbiased gradient estimate based on a sample t of the random variable T:

$$g(t, \theta) = f(t) \nabla_\theta \log q(t \mid \theta). \tag{15}$$

The estimate does not impose restrictions on the form of the function f, but requires an efficient algorithm for generating t and computing $\log q(t \mid \theta)$. The latter imposes additional restrictions on certain classes of discrete structured variables, such as distributions based on exponential families. In practice, the convergence of the algorithm can be hindered by the high variance of the estimate $g(t, \theta)$; as a result, the algorithm requires additional control variates to mitigate the gradient variance.

The reparameterization trick allows us to estimate the gradients in 13 under the assumption that the random variable T can be represented as $T = h(U, \theta)$ for a smooth f and a smooth with respect to the second argument h and some random variable U. As the name suggests, the gradient estimate is obtained by differentiating the expectation in a new parameterization

$$\nabla_\theta \mathbb{E}_T f(T) = \nabla_\theta \mathbb{E}_U f(h(U, \theta)) = \mathbb{E}_U \nabla_\theta f(h(U, \theta)), \tag{16}$$

giving an estimate that depends on the sample u of U as

$$g(u, \theta) = \nabla_\theta f(h(u, \theta)) = \left.\frac{\partial f}{\partial t}\right|_{t=h(u,\theta)} \frac{\partial h}{\partial \theta}. \tag{17}$$

Compared to estimate 15, the reparameterized estimate in practice has a lower variance, but imposes additional restrictions on f and T as it involves derivatives. In particular, the estimate 17 is not directly applicable to discrete variables, allowing estimation of gradients only for their continuous approximations.

3 Main Results

Below we cover the central results of the thesis.

3.1 General Methods

3.1.1 Permutation Prediction Based on Variational Relaxation

Our work [21] focuses on methods for approximate inference in the case when the structured hidden variable T is a random permutation. We consider variational distributions within the Plackett-Luce parametric distribution family.

Definition 3. A Plackett-Luce distribution with parameters $\theta_1, \ldots, \theta_n$ is a distribution on permutations with the probability of outcome $t \in S_n$ equal to

$$\mathbb{P}_T(T = t; \theta) = \prod_{i=1}^{n} \frac{\exp \theta_{t_i}}{\sum_{j=i}^{n} \exp \theta_{t_j}}. \tag{18}$$

Intuitively, the above formula corresponds to choosing n out of n elements without replacement, where the probability of choosing the i-th element is proportional to $\exp\theta_i$. The distribution is also of interest from the point of view of probabilistic relaxation of optimization problems. We replace the minimum over the function arguments with the minimum of the average function value over the distribution family parameters

$$\min_t f(t) \leq \min_\theta \mathbb{E}_T f(T), \tag{19}$$

where T has the Plackett-Luce distribution with parameters θ. This estimate smoothly depends on the distribution parameters. Importantly, it is possible to find a set of parameters that leads to an arbitrarily small gap in the above inequality. Indeed, when we scale the parameters $\theta' = \theta/\tau$ by the temperature τ approaching zero, the distribution tends to degenerate distribution. The distribution mode is the permutation that arranges θ in descending order, since such sorting delivers the maximum of each factor in formula 18. Therefore, if the sorting of the vector θ coincides with the optimal permutation of τ^*, temperature scaling lead to an arbitrarily small gap.

Explicit formula for the outcome probability 18 and generation using sampling without replacement allow using the REINFORCE [71] algorithm for approximate inference in the class of Plackett-Luce distributions. However, the default algorithm converges slowly due to high variance, so as part of our work, we adapted the RELAX [24] algorithm to obtain low variance gradient estimates.

Definition 4. Let a discrete random variable T be a function $T = H(Z)$ of a reparameterizable random variable Z with parameters θ. Then the estimate

$$g_{RELAX}(f) = [f(t) - c_\phi(\tilde{z})]\,\frac{\partial}{\partial\theta}\log\mathbb{P}_T(T=t;\theta) + \frac{\partial}{\partial\theta}c_\phi(z) - \frac{\partial}{\partial\theta}c_\phi(\tilde{z}) \tag{20}$$

for a realization z of a random variable Z, a discrete variable $t = H(z)$, and an independent realization $\tilde{z}$ of a conditional random variable $Z \mid T = t$ is an unbiased estimate of $\mathbb{E}_T f(T)$.

Initially, a similar estimate was proposed in [66], where $c_\phi(\cdot)$ was considered to be a smooth extension of function f to the domain of Z, containing the domain of T. In the paper [24], the authors proposed using an arbitrary $c_\phi(\cdot)$ (assuming differentiability with respect to the argument z and the parameters ϕ) while adjusting the parameters ϕ as it is optimized to reduce the variance. Both papers considered the case of a categorical distribution, while we generalized the method to the case of the Plackett-Luce distribution.

Our generalization is based on the equivalent definition of the Plackett-Luce distribution [73].

Definition 5. Let $Z_1, \ldots, Z_n$ be independent random variables with the Gumbel distribution with the corresponding parameters $\theta = (\theta_1, \ldots, \theta_n)$. Then the sorting of these random variables T has the Plackett-Luce distribution:

$$\mathbb{P}(z_{t_1} \geq \cdots \geq z_{t_n}; \theta) = \prod_{i=1}^{n} \frac{\exp\theta_{t_i}}{\sum_{j=i}^{n}\exp\theta_{t_j}}. \tag{21}$$

Thus, the random variable T can be represented as a deterministic function of Z, and in order to use the bound 20, it suffices to find a reparameterization for the conditional distribution $Z \mid T = t$. In our work, we propose an algorithm for reparameterization and efficient generation from this distribution:

Theorem 2. *Consider mutually independent realizations of the uniform distribution* $v_1, \ldots, v_n \sim U[0,1]$ *and realizations of the Gumbel distribution* $z_1, \ldots, z_n$ *with parameters* $\theta_1, \ldots,$ *theta*$_n$. *Then for the permutation* $t = \arg\operatorname{sort}(z_1, \ldots, z_n)$ *the vector* $\tilde{z} = (\tilde{z}_1, \ldots, \tilde{z}_n)$ *defined as*

$$\tilde{z}_{t_i} = \begin{cases} -\log(-\log(v_i)) & i = 1 \\ -\log\left(\frac{\log v_i}{\sum_{j=i}^{n} \exp\theta_{t_j}} + \exp(-\tilde{z}_{t_{i-1}})\right) & i \geq 2 \end{cases} \tag{22}$$

is a realization of the conditional distribution $Z \mid T = t$.

We studied the performance of the proposed method on the problem of finding a causal data structure, considering several problem settings. First, we considered synthetic data generated from the Structured Equation Model, [46]. To generate data, we chose a random directed acyclic graph $G = (E, V)$ with a weighted adjacency matrix $W \in \mathbb{R}^{n \times n}$, and then generated data $X \in \mathbb{R}^{n \times N}$ satisfying the equation

$$X = W^T X + \varepsilon, \tag{23}$$

where ε is homoscedastic Gaussian noise. This equation describes a linear dependence in which each component of X_i depends on the parents of the vertex i in the graph G, as well as on the random noise. The task was to restore the structure of the graph G from the data X.

To reduce the problem to the problem of inferring a permutation, we parameterized the desired adjacency matrix W based on topological sorting: $W = PAP^T$, where $A \in \mathbb{R}^{n \times n}$ was a strictly upper triangular matrix, and P was the permutation matrix for the topological sorting of the graph. For the chosen parameterization, we solved the problem

$$\min_{P \in \mathbb{P}} \min_{A \in \mathbb{A}} \frac{1}{2N} \|X - PAP^T X\}_F^2 + \lambda \|\operatorname{vec}(A)\|_1 = Q(P, A), \tag{24}$$

where $\mathbb{P}$ is the set of permutation matrices and $\mathbb{A}$ is the set of strictly upper triangular matrices.

To optimize over the set of permutations, we switched to the probabilistic relaxation

$$\min_{\theta} \mathbb{E}_T \min_{A \in \mathbb{A}} Q(P(T), A). \tag{25}$$

We compared our method with the previously proposed Gumbel-Sinkhorn [40] and URS [26] algorithms based on P permutation matrix relaxation. The table 1 shows the results of experiments for four families of graphs with 20 vertices. The algorithms proposed for comparison are significantly inferior to our approach both in terms of the quality of optimization of the objective function and in terms of the structural metrics SHD, SHD-CPDAG, and SID. We have also improved the Sinkhorn and URS algorithms by adding additional optimization constraints, obtaining comparable results. A full description of this experiment, as well as other experiments, can be found in the corresponding chapter of the thesis.

3.1.2 Learning Guarantees for Quadratic Surrogate Losses

In the paper [61] we analyzed surrogate loss functions with a specific focus on structured prediction. The main theoretical result of the work is a strengthened lower bound on the calibration function for a quadratic surrogate loss, which allows one to obtain non-trivial guarantees in the case when the surrogate loss is not consistent.

ER1				
	Val Q - Ql	SHD	SHD-CPDAG	SID
PL-RELAX	15.7±27.3	14.4±5.3	16.0±6.2	61.0±48.7
SINKHORN_{ECP}	10.4±8.7	15.8±4.7	17.0±6.0	84.8±56.3
URS_{ECP}	27.5±34.2	20.6±6.3	21.4±7.2	96.8±74.6
SINKHORN	1651.2±3050.4	24.0±6.1	25.0±6.7	131.2±76.5
GREEDY-SP	N/A	18.6±13.5	18.0±16.6	74.0±53.5
RANDOM	895.1±1270.3	37.8±5.2	38.8±4.9	146.8±79.9
SF1				
	Val Q - Q*	SHD	SHD-CPDAG	SID
PL-RELAX	-1.5±0.2	4.0±0.6	4.6±0.5	4.2±0.7
SINKHORN_{ECP}	1.9±4.3	6.6±2.2	6.6±2.4	10.4±5.0
URS_{ECP}	3.0±2.0	10.6±2.0	10.6±1.6	14.4±4.0
SINKHORN	38.3±26.2	19.0±0.0	19.0±0.0	35.0±2.4
URS	38.3±26.2	19.0±0.0	19.0±0.0	35.0±2.4
GREEDY-SP	N/A	2.0±1.4	0.0±0.0	7.0±5.1
RANDOM	94.0±36.4	36.2±2.6	36.6±2.3	48.6±14.7
ER4				
	Val Q - Q*	SHD	SHD-CPDAG	SID
PL-RELAX	468.8±208.4	71.0±5.9	72.6±3.9	289.6±9.1
SINKHORN_{ECP}	2519.0±3715.2	78.0±6.1	78.8±5.5	302.2±15.8
URS_{ECP}	1011.4±745.5	75.8±2.9	76.6±2.9	300.2±20.3
SINKHORN	126284.6±194386.3	88.8±6.0	91.0±5.7	330.0±14.1
GREEDY-SP	N/A	103.4±10.9	105.6±10.5	288.6±14.7
RANDOM	109891.2±74968.7	113.0±4.9	114.4±4.1	330.6±9.2
SF4				
	Val Q - Q	SHD	SHD-CPDAG	SID
PL-RELAX	-5.8±1.2	20.0±4.3	20.0±4.1	48.4±16.2
SINKHORN_{ECP}	-0.4±2.4	25.6±5.6	25.8±5.9	58.6±19.7
URS_{ECP}	8.5±11.8	30.2±5.8	30.6±5.2	72.2±25.0
SINKHORN	158.2±99.9	44.6±5.8	44.8±6.1	103.6±20.8
URS	140.7±140.6	42.0±5.4	42.8±5.1	89.8±20.4
GREEDY-SP	N/A	50.6±31.5	49.8±32.3	69.0±43.2
RANDOM	635.5±182.6	98.2±6.1	99.2±5.5	168.8±29.6

Table 1: Метрики для графов из 20 вершин

Following the notation introduced in Section 2.2, we introduce a quadratic surrogate loss function

$$\Phi_{quad}(f, y) := \frac{1}{2k}\|f + L(:, y)\|_2^2 = \frac{1}{2k}\sum_{\hat{y}\in\mathcal{Y}}\left(f_{\hat{y}}^2 + 2f_{\hat{y}}L(\hat{y}, y) + L(\hat{y}, y)^2 + L(\hat{y}, y)^2\right). \quad (26)$$

As noted above, the prediction f is often parameterized using an additional matrix $F : f(x) = Fg(x)$. Earlier Osokin et al. [44] obtained a calibration function lower bound under the assumption that the linear span $\mathcal{F}$ of the columns of the matrix F coincides with the linear span of the columns of the loss function.

Theorem 3. *For any loss matrix L, the corresponding quadratic surrogate Φ_{quad}, and the prediction space $\mathcal{F}$ containing the columns of the matrix L, the calibration function $H_{\Phi_{quad},L,\mathcal{F}}$ satisfies*

$$H_{\Phi_{quad},L,\mathcal{F}}(\varepsilon) \geq \frac{\varepsilon^2}{2k\max_{i\neq j}\|P_{\mathcal{F}}\Delta_{ij}\|_2^2} \geq \frac{\varepsilon^2}{4k}, \quad (27)$$

where $P_{\mathcal{F}}$ is the orthogonal projection operator onto $\mathcal{F}$ and the vector $\Delta_{ij} = e_i - e_j \in \mathbb{R}^k$, where e_c denotes c- th standard basis vector in $\mathbb{R}^k$.

The latter inequality is trivial and leads to the estimate obtained by Ciliberto et al. [14]. On the other hand, as $\mathcal{F} \subsetneq \mathbb{R}^k$ decreases, the projection norm $\|P_{\mathcal{F}}\Delta_{ij}\|_2^2$ drops, resulting in more accurate lower bounds for the calibration function. The minimum set of scores that satisfies the conditions of the theorem is $\mathcal{F} = \operatorname{span} L$.

In our work, we have relaxed the constraint $\operatorname{span} L \subset \mathcal{F}$, obtaining the following estimate.

Theorem 4. *For any loss matrix L, the corresponding quadratic surrogate Φ_{quad}, and the prediction space $\mathcal{F}$, the calibration function $H_{\Phi_{quad},L,\mathcal{F}}$ satisfies*

$$H_{\Phi_{quad},L,\mathcal{F}}(\varepsilon) \geq \min_{i\neq j}\max_{v\geq 0}\frac{(\varepsilon v - \xi_{ij}(v))_+^2}{2k\|P_{\mathcal{F}}\Delta_{ij}\|_2^2}, \text{ where } \xi_{ij}(v) := \|L^T(vI_k - P_{\mathcal{F}})\Delta_{ij}\|_\infty, \quad (28)$$

the operator $P_{\mathcal{F}}$ defines an orthogonal projection onto $\mathcal{F}$, the function $(x)_+^2 := [x > 0]x^2$ defines the right branch of the parabola and $\Delta_{ij} := e_i - e_j \in \mathbb{R}^k$, where e_c denotes the cth standard basis vector in $\mathbb{R}^k$.

Assuming $v = 1$ in the estimate introduced above, we can also obtain a simplified expression

$$H_{\Phi_{quad},L,\mathcal{F}}(\varepsilon) \geq \min_{i\neq j}\frac{(\varepsilon - \xi_{ij})_+^2}{2k\|P_{\mathcal{F}}\Delta_{ij}\|_2^2}, \text{ where } \xi_{ij} := \|L^T(I_k - P_{\mathcal{F}})\Delta_{ij}\|_\infty. \quad (29)$$

Importantly, for $\operatorname{span} L \subset \mathcal{F}$ the new lower bound 28 is at least as tight as the old one 27. Indeed, the expression inside coincides with the old estimate for $v = 1$, but can deliver a tighter bound when $v \neq 1$. In the case when $\mathcal{F}$ does not contain the columns of $\mathcal{F}$, the lower one will be the envelope of the family of curves with parameter v. Each of the curves is the right branch of a parabola shifted to the right. Near zero, the lower bound is zero due to the inconsistency of the surrogate when $\mathcal{F} \not\subset \operatorname{span} L$. The leftmost point with positive bound is equal to $\eta = \frac{\xi_{ij}(v)}{v}$ and determines the level of consistency of the surrogate in a broad sense.

In addition to deriving a general estimate, we calculate the constants in the inequality and analyze several popular loss functions. As an illustration, we present the loss function

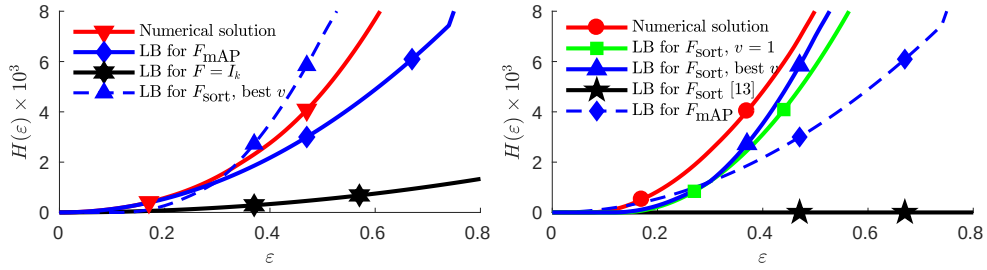

Figure 1: Left: consistent calibration function for F_{mAP}; right: inconsistent calibration function for F_{sort}

mAP (mean average precision) used in ranking problems [10, 9, 49]. In this case, the model prediction $\sigma \in \hat{\mathcal{Y}} = S_r$ is a permutation of r elements, and the labels $y \in \mathcal{Y} = \{0,1\}^r$ are binary vectors of length r. The loss function $L_{mAP}(\sigma, y)$ averages the ranking accuracy for different recall levels:

$$L_{mAP}(\sigma, y) := 1 - \frac{1}{|y|} \sum_{p:y_o=1}^{r} \frac{1}{\sigma(p)} \sum_{q=1}^{\sigma(p)} y_{\sigma^{-1}(q)} = 1 - \sum_{p=1}^{r} \sum_{q=1}^{p} \frac{1}{\max(\sigma(p), \sigma(q))} \frac{y_p y_q}{|y|}. \quad (30)$$

Above, the norm of a binary vector is $|y| = \sum_{p=1}^{r} y_p$. The second expression for the loss matrix leads to two natural definitions of $\mathcal{F}$, which we present below. For the first parameterization, we define $\mathcal{F}_{mAP} = \operatorname{span} F_{mAP}$ in terms of the linear span of the columns of the matrix $F_{mAP} \in \mathbb{R}^{r! \times \frac{1}{2}r(r+1)}$ with elements $(F_{mAP})_{\sigma,pq} := \frac{1}{\max(\sigma(p),\sigma(q))}$. It follows from the definition of L_{mAP} that $\operatorname{span} L_{mAP} = \operatorname{span} F_{mAP}$, and the quadratic surrogate loss function is consistent. On the other hand, the derivation to this model reduces to the integer quadratic programming problem $\max_{\sigma \in S_r} (F_{mAP}\theta)_\sigma$, which is NP-hard. For the second parameterization, we define $\mathcal{F}_{sort} = \operatorname{span} F_{sort}$ as the linear span of the matrix $F_{sort} \in \mathbb{R}^{r! \times R}$ with elements $(F_{sort})_{\sigma,p} := \frac{1}{\sigma(p)}$. In this parameterization, inference task $\max_{\sigma \in S_r} (F_{sort}\theta)_\sigma$ is equivalent to sorting the elements of θ, which makes the second parameterization preferable to the first. On the other hand, $\mathcal{F}_{sort}$ does not contain the columns of the matrix L_{mAP}, which makes the quadratic surrogate inconsistent.

The figure 1 shows the graphs of the estimates described above for the loss function L_{mAP}. Due to the inconsistency of the surrogate loss function, the graph for F_{sort} is zero up to a certain $\varepsilon > 0$. At the same time, for some values of ε, the calibration function lower bound for F_{sort} turns out to be higher than the calibration function lower bound for F_{mAP}. In practice, this means that for lower optimization precision, our bound provides stronger learning guarantees for the parameterization F_{sort} with the efficient inference algorithm.

3.2 Applications

3.2.1 Structured Priors for Convolutional Neural Network Kernels

Our work [1] proposes to interpret the parameters of a convolutional neural network as a structured latent variable. Compared to basic Bayesian neural networks, the structured prior distribution of network parameters takes into account dependencies between individual weights of convolutional filters. In experiments, the proposed modification improved the classification quality in a setup with a limited training set, allowed to speed up network training, and allowed to extract low-dimensional data representations without additional training.

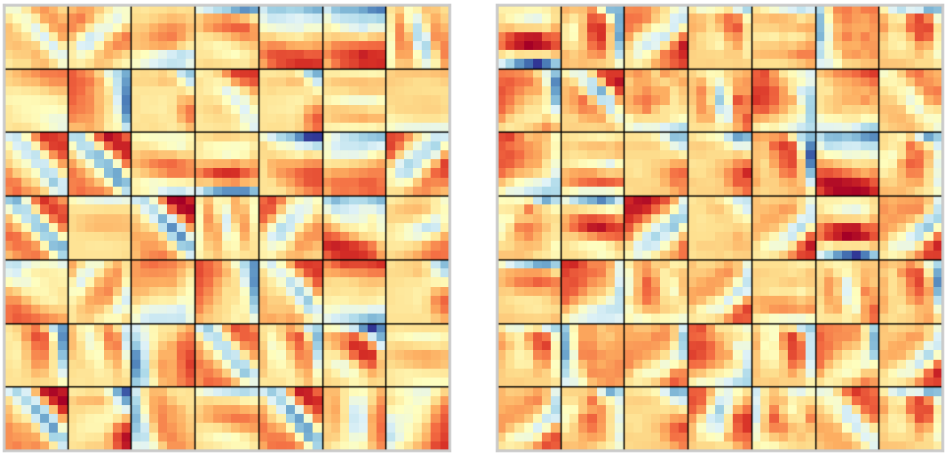

Figure 2: Left: Trained convolutional network filters. Right: filters obtained from approximation.

Bayesian neural network is a discriminative model at the intersection of Bayesian methods of machine learning and deep learning. The joint distribution

$$p(y^1, \dots, y^n, \theta \mid x^1, \dots x^n) = \left[\prod_{i=1}^{n} p(y^i \mid x^i, \theta)\right] p(\theta) \tag{31}$$

on class labels $y^1, \dots, y^n$ and model weights θ typically consists of a set of independent distributions for each individual weight $p(\theta) = \prod_j p(\theta_j)$ and the likelihood of the label $p(y^i \mid x^i, \theta)$ given the input object x^i and the weights θ. For prediction the model combines the weights posterior distribution $p(\theta \mid \{x^i, y^i\}_{i=1}^n)$ the label distribution $p(y^i \mid x^i, \theta)$ into a posterior predictive distribution:

$$p(y_{test} \mid \{x^i, y^i\}_{i=1}^n, x_{test}) = \int p(y_{test} \mid x_{test},\ theta) p(\theta \mid \{x^n, y^n\}_{n=1}^N) \mathrm{d}\theta. \tag{32}$$

In practice, the posterior distribution is approximated via variational inference, i.e. by solving the problem

$$\max_{\phi} \left[\mathbb{E}_{\Theta} \log p(y^1, \dots, y^n \mid \Theta, x^1, \dots, x^n) - KL(q(\Theta; \phi) || p(\Theta))\right], \tag{33}$$

where the expectation is taken with respect to the variational distribution with density $q(\theta; \phi)$ and parameters ϕ. The assumption of the independence of the parameters in the prior distribution simplifies the parameterization of the model. However, the weights of the convolutional filters of a trained neural network do not behave like independent random variables. Qualitatively, the weights smoothly change depending on the location in the filter (Figure 2 illustrates the argument). Moreover, in applications, the trained parameters of the convolutional network can be used in a new task on a similar domain [74, 57]. For example, in the case of images, ImageNet-trained convolutional networks can be adapted to other computer vision tasks.

We consider the distribution of convolutional filters on a certain domain. We propose an empirical approximation to the distribution. In particular, we train several convolutional networks on an auxiliary task in the same domain. The auxiliary task must be representative of the given domain: training examples must be diverse, and the network representations must be sufficiently informative. In this work, we considered image classification, the auxiliary task was a classification task with a different training set and

a different set of labels. Having trained several convolutional networks, we can build an empirical approximation of filter distribution. However, there are two problems with empirical approximation. First, to work with the distribution, it is necessary to store many convolutional networks in memory. Secondly, the density required to calculate the objective function of the Bayesian neural network is not available for the approximation. Therefore, we propose to approximate the distribution of filters using an auxiliary generative model based on a variational auto-encoder.

When training, we propose to replace the prior distribution of $p(W)$ with an estimate obtained on the basis of a variational auto-encoder. Thus, we arrive at a lower bound on the marginal likelihood

$$\log p(\{y^i\}_{i=1}^n \mid \{x^i\}_{i=1}^n) \geq \mathbb{E}_\Theta \left[\log p(\{y^i\}_{i=1}^n \mid \Theta, \{x^i\}_{i=1}^n)\right. \tag{34}$$

$$+ \mathbb{E}_Z \log \frac{p(\Theta \mid Z; \chi) p(Z)}{r(Z \mid \Theta; \psi)} \tag{35}$$

$$\left. - \log q(\Theta; \phi)\right], \tag{36}$$

where $q(\Theta; \phi)$ is a variational approximation of the network parameters, the distributions of $p(\Theta \mid Z; \chi)$ and $r(Z \mid \Theta; \psi)$ are determined by the variational auto-encoder , and the expectation with respect to the random vector Z is calculated with respect to the distribution $r(Z \mid \Theta; \psi)$. When training a Bayesian neural network, we will use this estimate as an objective function. The first term in the estimate corresponds to the standard cross-entropy loss function, and the second term pulls the approximate posterior distribution $q(\Theta; \phi)$ towards the empirical prior distribution of the convolutional network parameters for the given domain $p(\Theta)$.

To evaluate the proposed approach, we conducted a series of experiments that evaluated the learning ability with limited training data, the representations the network obtains after the initialization from the prior distribution, and the learning time depending on the weight prior distribution. Here we restrict ourselves to the first experiment, a detailed description of the others can be found in the corresponding chapter.

While studying Bayesian network training with limited data, we considered the classification problem on MNIST and CIFAR-10. As a model, we took convolutional networks consisting of several convolutional layers along with several fully connected output layers. For convolutional layers, we trained the prior distribution on NotMNIST and CIFAR-100 data. We trained the fully connected layers with backpropagation without resorting to variational inference. For comparison, we considered the prior distributions for network parameters common in the literature: the Gaussian distribution and the log-uniform distribution, among which the latter guarantees the invariance of the prior distribution to the scale of the parameters.

For the three prior distribution families we trained classifiers with varying sizes of training data. As Figure 3 shows, the network with the proposed prior distribution performs better. On the MNIST data, the difference in quality disappears when the training sample size is sufficient. On CIFAR-10 data, the quality is uniformly higher. We assume that the difference can be explained by the simplicity of the classification task on the MNIST data: thousands of examples are enough to extract the necessary information from the data.

3.2.2 Bayesian Estimation of Multiple Access Channel Configuration

Probabilistic Model. In the paper [60], we consider the problem of estimating the parameters of a multi-user communication channel. To establish a connection on dedicated

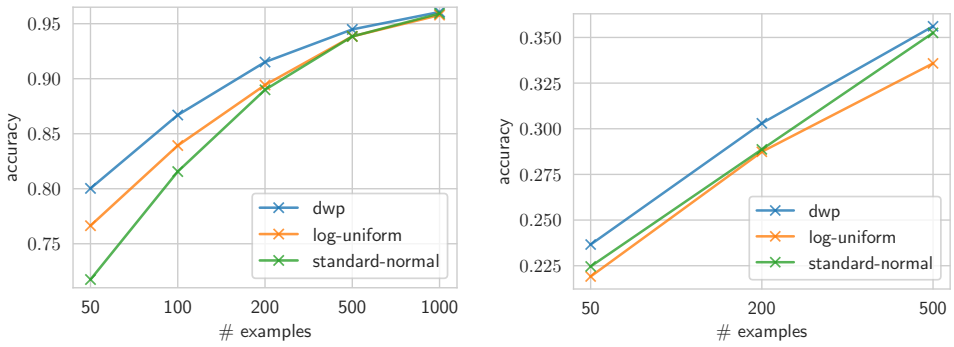

Figure 3: Classification quality depending on the size of the training sample. Left: MNIST, right: CIFAR-10.

frequencies, users send special code signals, which are then received and processed by a cellular communication station. The communication channel is multi-user, so some frequencies can be occupied by several users and the system must be able to detect users by receiving a superposition of the sent code signals. In practice, to simplify the task, it is assumed that there are quite a few users.

In fact, the problem can be interpreted as a structured prediction problem, where, based on the received signals, it is necessary to choose a sparse binary vector with a block structure. Standard solutions are based on modifications of compressed sensing algorithms. The work [70] proposed an approach based on Bayesian linear regression. Bayesian linear regression allows finding sparse solutions to linear systems of equations, which is required in this problem. In our work, we adapted the standard Bayesian linear regression model to the specifics of the problem, taking into account the block structure of the desired solution, and also proposed a faster algorithm for solving the task.

Mathematical model of the communication process is a system of linear equations

$$y = \kappa\theta + z, \tag{37}$$

where the vector y corresponds to the received signal, the matrix κ is fixed and specified by the communication protocol, the vector θ is unknown and describes the channel configuration, and z models the noise that occurs during signal transmission. As a noise model, we used a Gaussian distribution with a known variance ρ. In addition, the vector θ has a block structure

$$\theta = (c_{11}t_1, \ldots, c_{1Q}t_1, c_{21}t_2, \ldots, c_{2Q}t_2, \ldots, c_{N1}t_N, \ldots, c_{NQ}t_N), \tag{38}$$

where the binary variables $t_1, \ldots, t_N \in \{0, 1\}$ are equal to one if the user is active, and the values $c_{11}, \ldots, c_{NQ} \in \mathbb{R}$ reflect the physical parameters of the communication channel. Within this model, we are primarily interested in recovering the $t_1, \ldots, t_N$ values that indicate active users in the channel. In addition, we are also interested in estimating the vector θ, since it contains signal fading parameter.

To solve the problem, we consider a Bayesian linear regression model with the following joint distribution

$$p(y, \theta; \rho, \gamma) = p(y \mid \theta; \rho)p(\theta; \gamma) \tag{39}$$

$$p(Y = y \mid \Theta = \theta; \rho) = \mathcal{N}(y \mid \kappa\theta; \rho I) \tag{40}$$

$$p(\Theta = \theta; \gamma) = \mathcal{N}(\theta \mid 0, \mathrm{diag}(\underbrace{\gamma_1, \ldots, \gamma_1}_{Q}, \ldots,\ underbrace\gamma_N, \ldots, \gamma_{NQ})). \tag{41}$$

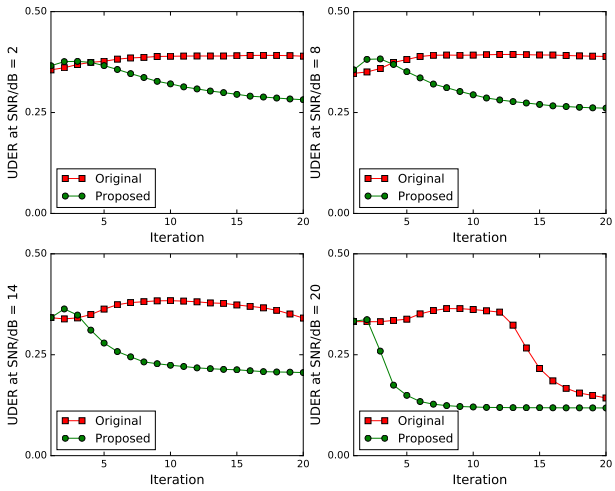

Figure 4: Dependence of the UDER (user detection error) on the number of iterations of the EM algorithm

The density $p(Y = y \mid \Theta = \theta; \rho)$ specifies the observations likelihood, and $p(\Theta = \theta; \gamma)$ specifies the prior distribution with the block structure. Note that in the previously proposed works, the basic regression model was used, which did not take into account the block structure of the prior distribution.

To estimate the channel configuration, we maximize the evidence $p(y; \rho, \gamma)$ with respect to the prior distribution parameters γ. Similarly to [70], we use the EM-algorithm for the derivation, alternately estimating the posterior distribution $p(\theta \mid y, \rho, \gamma)$ at the E-step and maximizing the evidence estimate with respect to γ at the M-step. At the M-step, we use the iterative scheme proposed in [64]. For our problem, the scheme improved the inference speed in model experiments compared to the previously considered schemes [70].

Simulation results. We ran a simulation to evaluate the performance of the proposed scheme. We compared reconstruction error of a model with a custom probabilistic model and the improved iterative scheme against the solution proposed in [70]. We used the Rayleigh fading model to model the signal amplitude, considered a channel with 6 active users out of $N = 36$, each using $Q = 5$ frequencies. We used Zadov-Chu sequences of length 20 to construct the codebook matrix κ. The graph 4 shows the dependence of the average proportion of incorrectly identified users $\text{UDER} = \mathbb{E}\left[frac\sum_{n=1}^{N}[\hat{a}_n \neq a_n N\right]$ on the number of iterations of the EM-algorithm for different signal-to-noise ratio levels in the communication channel. In all four cases, the proposed scheme converges faster than the original scheme. Moreover, for high noise levels, the original scheme does not converge on average. The graph 5 shows the dependence of the average Θ estimation error on the number of iterations of the EM-algorithm. As in the previous experiment, the proposed scheme shows the best convergence. It is noteworthy that in terms of the MSE metric, the original scheme achieves comparable results even for high noise levels.

3.3 Pre-processing of Geological Survey Data with Hidden Markov Chains

Probabilistic model. The last chapter focuses on the analysis of geological survey data. In our work [59], we adapt the hidden Markov chain to the task of pre-processing and imputation for missing data in well logging. The Hidden Markov Chain is one of the classic probabilistic models with a latent structured variable: the hidden variable is given by a Markov chain with discrete states, the observations are independent under given

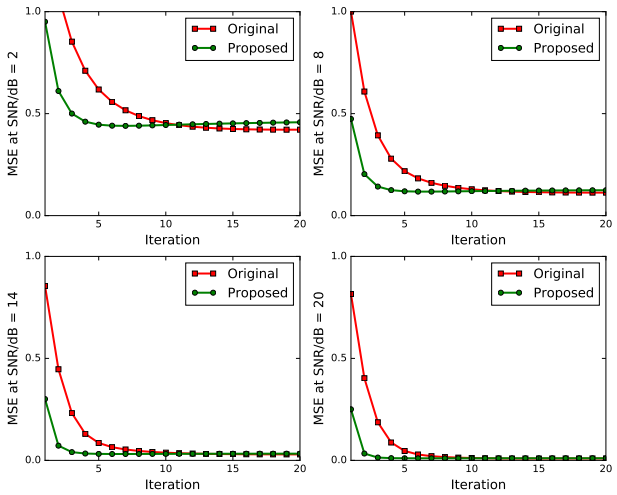

Figure 5: Dependence of the Θ communication channel parameters estimation error on the number of iterations of the EM algorithm

the hidden variable, and the EM-algorithm is used to tune the parameters and infer the hidden variable.

In the first stages, the goal of a geological survey is to build a model of the deposit. The model is based on number wells drilled on the territory of the field, the survey data is collected with a number of sensors that are lowered into each well. As you move deeper, the sensors read different physical characteristics of the well depending on the depth. The acquired data form sequences referred to as "logs". Then, based on this data, an expert petrophysicist labels segments of wells that are of interest from the viewpoint of field development. To produce the labels the expert also performs data alignment, additional calibration and anomaly search.

The goal of this study is to automate data processing steps of an expert petrophysicist. The results of the work of experts accumulated over many years make it possible to solve well labeling as a supervised learning task, however, the predictions of experts are subjective and may not provide an insufficiently reliable training signal. Therefore, we considered an unsupervised learning setup that could provide consistent predictions across all the wells.

Next we describe the proposed probabilistic model. Let $x^1, \ldots, x^K$ be logs for for K wells, $x^k \in \mathbb{R}^{l_k \times d}$. For each well at a given depth level, the sensor reading is primarily determined by the soil characteristics. We assume that soil characteristics can be described by a sequence of m states of a Markov chain. To define the Markov chain we introduce random vectors $T^1, \ldots, T^K$, $T^k_l \in \{1, \ldots, m\}, l = 1, \ldots, L^k k$, initial distribution $P(T^k_1 = t; \pi) \propto \pi_k, \pi \in \mathbb{R}^m_+$ and consecutive pair distributions $P(T^k_l = t \mid T^k_{l-1} = s; \tau) \propto \tau_{ts}, \tau \in \mathbb{R}^{m^2}_+$. The dependence of the elements of the chain allows to promote identical states for adjacent segments. Continuous segments of the chain with a constant latent state correspond to homogeneous sections of the well, along which soil characteristics do not change. In practice, soil characteristics are unknown, so the Markov chain acts as a latent variable in the model. However, we know the sensor readings in the logs. We assumed that for each type of soil, the sensor readings follows the multivariate Gaussian distribution $p(x^k_l \mid T^k_l = t; \mu, \Sigma) = \mathcal{N}(x^k_l \mid \mu_t, \Sigma_t), \mu \in \mathbb{R}^{m \times d}, \Sigma \in \mathbb{R}^{m \times d^2}$.

Besides that, sensor readings are affected by instrument calibration prior to recording. Assuming that sensor calibration can be represented as a linear transformation of the

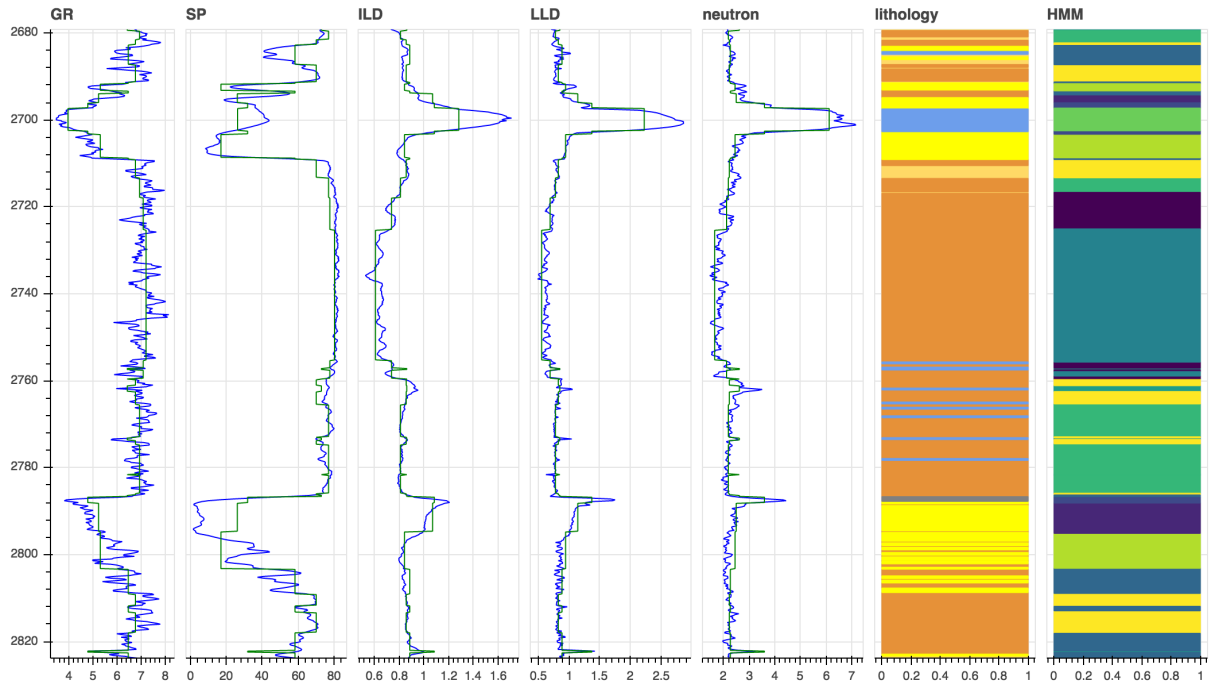

Figure 6: An illustration of how the model works for one well. Left: Example of observed logs (blue) and their approximation (green); right: expert labels compared to the found states of the Markov chain.

readings $x_l^k = \alpha_k \odot \hat{x}_l^k + \beta_k$ for a given observation x_l^k and a calibrated observation $\hat{x}_l^k$, we introduce additional calibration parameters $\alpha \in \mathbb{R}_*^{K\times d}, \beta \in \mathbb{R}^{K\times d}$ for each well. For the parameters $\Theta = (\pi, \tau, \alpha, \beta)$, the final observational model is

$$p(x^i, t^i{}_{i=1}^K; \Theta) = \tag{42}$$

$$\prod_{k=1}^{K}[(\pi_{t_1^k} \prod_{l=2}^{L_k} \tau_{t_l^k t_{l-1}^k})\times \tag{43}$$

$$\prod_{l=1}^{L_k} \mathcal{N}(x_l^k \mid \alpha_k \odot \mu_{t_l^k} + \beta_k, \mathrm{diag}(\alpha_k)\Sigma_{t_l^k}\,\mathrm{diag}(\alpha_k))]. \tag{44}$$

To tune the model parameters, we use the Baum-Welch algorithm to maximize the evidence lower bound of the model. We maximize the lower bound with stochastic gradient descent, starting from a hand-crafted initialization to avoid local optima. Since the observations in the model follow the Gaussian distribution, we could incorporate data with gaps using marginal distributions as a likelihood, without taking into account the missing data. We used the Viterbi algorithm to predict the hidden states of the circuit. Below we present the results of the model operation on synthetic fields, as well as on the Priobskoye field [3].

Empirical results. We started with a synthetic field, for which we both have measurements for wells, and the ground truth labels for soil types. As a result, we were able to qualitatively compare the hidden states of the Markov chain with the ground truth labels, thus eliminating the factor of subjective data interpretation by an expert. On the left, the graph 6 contains the logs (blue lines) as well as the predictions of our model (green line). While we were able to accurately replicate the behavior of the logs, we did not get a one-to-one correspondence between latent states and soil types. The ground truth

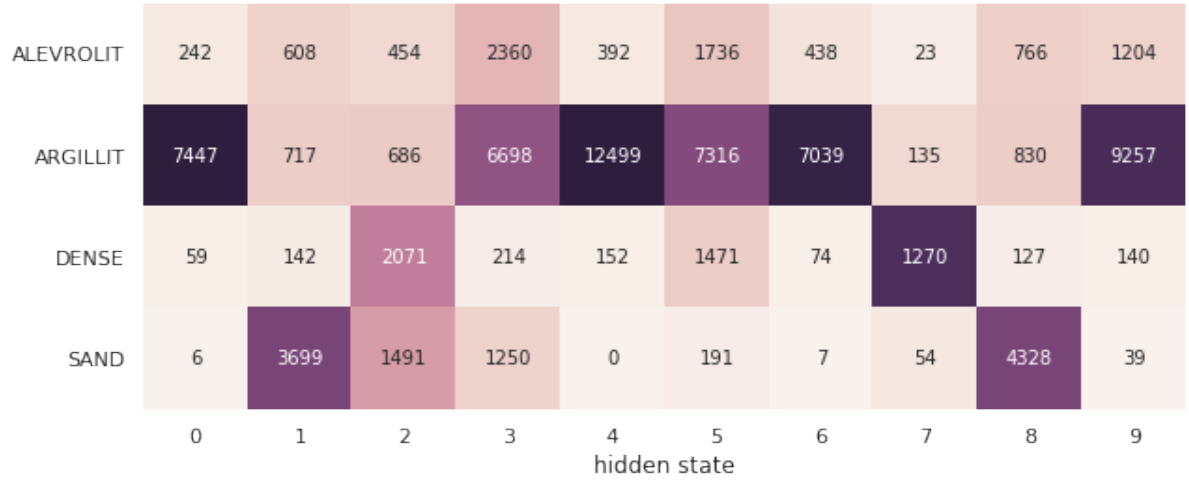

Figure 7: Correspondence between soil types and hidden states of the Markov chain

labels and hidden states of the well is shown in the graph 6 on the right. The selected number of latent states exceeded the number of soil types: increasing the number of latent states improves the approximation of logs, but makes the latent states less interpretable. On the graph 7 we have shown the correspondence between latent states and soil types throughout the field. Most of the latent states correspond to the argillite prevailing in the deposit. The model was also able to separate tight rocks and sandstones, but none of the hidden states correspond to siltstone.

We then applied the model to pre-process the data at the Priobskoye field. The base model predicted reservoir layers (layers of interest in terms of oil production) using a binary classification based on a recurrent neural network [3]. In the base model, instrument readings were standardized to account for miscalibration. We, in turn, calibrated the reading using the calibration parameters α, β obtained with a hidden Markov model instead of standardizing the data. The new pre-processing algorithm did not give a significant improvement in the quality of the prediction, increasing the F1 score from 0.72 to 0.74.

Next, we used the model to fill the gaps in the data. We considered test wells for which there are no ILD (deep induction log) and LLD (lateral log) log values in the sample. We then compared two gap recovery strategies: replacing the log with the average across the field, and our approach of restoring the log from the rest of the logs using a Markov chain. The proposed solution improved the prediction quality for the considered wells from F1=0.37 to F1=0.56. Thus, the proposed approach allows us to improve the quality of finding reservoir layers due to joint calibration and recovery of gaps in the data.

4 Conclusion

The results described above cover various aspects of structured prediction, including theoretical analysis of the standard structured prediction setup, models with latent structured variables based on a probabilistic approach, as well as applications of the described solutions to real problems. In conclusion, we briefly summarize the presented results.

1. We proposed a permutation optimization method based on probabilistic relaxation and the REINFORCE algorithm; we developed control variates to improve the con-

vergence of the method. We evaluated the method on the problem of identifying causal links in data, where the topological sorting of a directed acyclic link graph acts as a structured variable. The proposed method significantly improved structure reconstruction metrics in comparison with relaxation-based gradient optimization methods. Since the considered optimization method does not introduce additional assumptions about the objective function and is actually a zero-order optimization method, in the future it can also be used for direct optimization of the objective function in structured prediction problems (without using auxiliary surrogate loss functions), as well as for amortized inference of permutations.

2. In a supervised structured prediction setup, we analyzed a training approach based on quadratic surrogate loss functions. In particular, we considered the case of inconsistent surrogate loss function, for which we obtained guarantees for the accuracy of the expected risk optimization. Assuming a fixed number of training samples and early optimization stopping, the analysis delivers tighter upper bounds on the expected risk values. From a practical point of view, the above formulation also leads to more efficient inference algorithms.

3. We considered a number of applications based on a probabilistic approach to structured prediction. First, we applied the variational auto-encoder model to infer the parameters of a convolutional neural network based on a priori knowledge about the network parameter distribution for a given domain. In this case, the parameters of the convolutional filters act as a latent structured variable, and the proposed approach improves the prediction accuracy of Bayesian neural networks for similar domains. Second, we considered the task of estimating the parameters of a multi-user communication channel, where the subset of active users acts as a hidden structured variable. In this case, we proposed an improved probabilistic model to estimate the structured variable, and accelerated the inference algorithm. Finally, we proposed a probabilistic model based on hidden Markov chains to model and interpret geophysical survey data. The proposed model uses structured variables, in this case the Markov chain hidden states, to infer and cluster the physical characteristics of the wells while modeling the joint distribution of these characteristics. Based on the reconstructed hidden states, we proposed an approach to data imputation and anomaly detection.

References

[1] Andrei Atanov, Arsenii Ashukha, Kirill Struminsky, Dmitriy Vetrov, and Max Welling. The deep weight prior. In *International Conference on Learning Representations*, 2018.

[2] Peter L Bartlett, Michael I Jordan, and Jon D McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

[3] Boris Belozerov, Nikita Bukhanov, Dmitry Egorov, Adel Zakirov, Oksana Osmonalieva, Maria Golitsyna, Alexander Reshytko, Artyom Semenikhin, Evgeny Shindin, and Vladimir Lipets. Automatic well log analysis across priobskoe field using machine learning methods. In *SPE Russian Petroleum Technology Conference*. OnePetro, 2018.

[4] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

[5] Quentin Berthet, Mathieu Blondel, Olivier Teboul, Marco Cuturi, Jean-Philippe Vert, and Francis Bach. Learning with differentiable pertubed optimizers. *Advances in neural information processing systems*, 33:9508–9519, 2020.

[6] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[7] Céline Brouard, Marie Szafranski, and Florence d'Alché Buc. Input output kernel regression: Supervised and semi-supervised structured output prediction with operator-valued kernels. *Journal of Machine Learning Research*, 17:np, 2016.

[8] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[9] David Buffoni, Clément Calauzenes, Patrick Gallinari, and Nicolas Usunier. Learning scoring functions with order-preserving losses and standardized supervision. In *ICML*, 2011.

[10] Clément Calauzènes, Nicolas Usunier, and Patrick Gallinari. On the (non-) existence of convex, calibrated surrogate losses for ranking. In *NIPS*, 2012.

[11] Jianbo Chen, Le Song, Martin Wainwright, and Michael Jordan. Learning to explain: An information-theoretic perspective on model interpretation. In *International Conference on Machine Learning*, pages 883–892. PMLR, 2018.

[12] Justin Chiu and Alexander M Rush. Scaling hidden markov language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1341–1349, 2020.

[13] Anna Choromanska, Alekh Agarwal, and John Langford. Extreme multi class classification. In *NIPS Workshop: eXtreme Classification, submitted*, volume 1, pages 2–1, 2013.

[14] Carlo Ciliberto, Lorenzo Rosasco, and Alessandro Rudi. A consistent regularization approach for structured prediction. *Advances in neural information processing systems*, 29:4412–4420, 2016.

[15] Shay B Cohen and Noah A Smith. Joint morphological and syntactic disambiguation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 208–217, 2007.

[16] Caio Corro and Ivan Titov. Differentiable perturb-and-parse: Semi-supervised parsing with a structured variational autoencoder. *arXiv preprint arXiv:1807.09875*, 2018.

[17] Corinna Cortes, Vitaly Kuznetsov, Mehryar Mohri, and Scott Yang. Structured prediction theory based on factor graph complexity. *Advances in Neural Information Processing Systems*, 29, 2016.

[18] David Roxbee Cox and David Victor Hinkley. *Theoretical statistics*. CRC Press, 1979.

[19] Koby Crammer and Yoram Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of machine learning research*, 2(Dec):265–292, 2001.

[20] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[21] Artyom Gadetsky, Kirill Struminsky, Christopher Robinson, Novi Quadrianto, and Dmitry Vetrov. Low-variance black-box gradient estimates for the plackett-luce distribution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10126–10135, 2020.

[22] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. In *International conference on machine learning*, pages 1243–1252. PMLR, 2017.

[23] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[24] Will Grathwohl, Dami Choi, Yuhuai Wu, Geoff Roeder, and David Duvenaud. Backpropagation through the void: Optimizing control variates for black-box gradient estimation. In *International Conference on Learning Representations*, 2018.

[25] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376, 2006.

[26] Aditya Grover, Eric Wang, Aaron Zweig, and Stefano Ermon. Stochastic optimization of sorting networks via continuous relaxations. In *International Conference on Learning Representations*, 2018.

[27] Jiatao Gu, Qi Liu, and Kyunghyun Cho. Insertion-based decoding with automatically inferred generation order. *Transactions of the Association for Computational Linguistics*, 7:661–676, 2019.

[28] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[29] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparametrization with gumble-softmax. In *International Conference on Learning Representations (ICLR 2017)*. OpenReview. net, 2017.

[30] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[31] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM computing surveys (CSUR)*, 54(10s):1–41, 2022.

[32] Yoon Kim, Sam Wiseman, and Alexander M Rush. A tutorial on deep latent variable models of natural language. *arXiv preprint arXiv:1812.06834*, 2018.

[33] Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. *Advances in neural information processing systems*, 27, 2014.

[34] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[35] Yi Lin. A note on margin-based loss functions in classification. *Statistics & probability letters*, 68(1):73–82, 2004.

[36] Ben London, Bert Huang, and Lise Getoor. Stability and generalization in structured prediction. *The Journal of Machine Learning Research*, 17(1):7808–7859, 2016.

[37] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

[38] C Maddison, A Mnih, and Y Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *Proceedings of the international conference on learning Representations*. International Conference on Learning Representations, 2017.

[39] André FT Martins, Tsvetomila Mihaylova, Nikita Nangia, and Vlad Niculae. Latent structure models for natural language processing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 1–5, 2019.

[40] Gonzalo Mena, David Belanger, Scott Linderman, and Jasper Snoek. Learning latent permutations with gumbel-sinkhorn networks. In *International Conference on Learning Representations*, 2018.

[41] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[42] Alex Nowak-Vila, Francis Bach, and Alessandro Rudi. A general theory for structured prediction with smooth convex surrogates. *arXiv preprint arXiv:1902.01958*, 2019.

[43] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[44] Anton Osokin, Francis R Bach, and Simon Lacoste-Julien. On structured prediction theory with calibrated convex surrogate losses. In *NIPS*, 2017.

[45] Max Benedikt Paulus, Dami Choi, Daniel Tarlow, Andreas Krause, and Chris J Maddison. Gradient estimation with stochastic softmax tricks. In *NeurIPS 2020*, 2020.

[46] Judea Pearl. *Causality*. Cambridge university press, 2009.

[47] Fabian Pedregosa, Francis Bach, and Alexandre Gramfort. On the consistency of ordinal regression methods. *Journal of Machine Learning Research*, 18:1–35, 2017.

[48] Lawrence Rabiner and Biinghwang Juang. An introduction to hidden markov models. *ieee assp magazine*, 3(1):4–16, 1986.

[49] Harish G Ramaswamy, Shivani Agarwal, and Ambuj Tewari. Convex calibrated surrogates for low-rank loss matrices with applications to subset ranking losses. In *NIPS*, 2013.

[50] D Raj Reddy et al. Speech understanding systems: A summary of results of the five-year research effort. *Department of Computer Science. Camegie-Mell University, Pittsburgh, PA*, 17:138, 1977.

[51] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[52] Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics*, 8:842–866, 2020.

[53] BTCGD Roller, C Taskar, and D Guestrin. Max-margin markov networks. *Advances in neural information processing systems*, 16:25, 2004.

[54] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[55] Yasubumi Sakakibara, Michael Brown, Richard Hughey, I Saira Mian, Kimmen Sjölander, Rebecca C Underwood, and David Haussler. Stochastic context-free grammers for trna modeling. *Nucleic acids research*, 22(23):5112–5120, 1994.

[56] Iqbal H Sarker. Machine learning: Algorithms, real-world applications and research directions. *SN Computer Science*, 2(3):1–21, 2021.

[57] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 806–813, 2014.

[58] Noah A Smith. Linguistic structure prediction. *Synthesis lectures on human language technologies*, 4(2):1–274, 2011.

[59] K Struminskiy, A Klenitskiy, A Reshytko, D Egorov, A Shchepetnov, A Sabirov, D Vetrov, A Semenikhin, O Osmonalieva, and B Belozerov. Well log data standardization, imputation and anomaly detection using hidden markov models. In *Petroleum Geostatistics 2019*, volume 2019, pages 1–5. European Association of Geoscientists & Engineers, 2019.

[60] Kirill Struminsky, Stanislav Kruglik, Dmitry Vetrov, and Ivan Oseledets. A new approach for sparse bayesian channel estimation in scma uplink systems. In *2016 8th International Conference on Wireless Communications & Signal Processing (WCSP)*, pages 1–5. IEEE, 2016.

[61] Kirill Struminsky, Simon Lacoste-Julien, and Anton Osokin. Quantifying learning guarantees for convex but inconsistent surrogates. In *NeurIPS*, 2018.

[62] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014.

[63] Ben Taskar, Carlos Guestrin, and Daphne Koller. Max-margin markov networks. *Advances in neural information processing systems*, 16, 2003.

[64] Michael E Tipping. Sparse bayesian learning and the relevance vector machine. *Journal of machine learning research*, 1(Jun):211–244, 2001.

[65] Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, Yasemin Altun, and Yoram Singer. Large margin methods for structured and interdependent output variables. *Journal of machine learning research*, 6(9), 2005.

[66] George Tucker, Andriy Mnih, Chris J Maddison, Dieterich Lawson, and Jascha Sohl-Dickstein. Rebar: Low-variance, unbiased gradient estimates for discrete latent variable models. *arXiv preprint arXiv:1703.07370*, 2017.

[67] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[68] Paul Voigt and Axel Von dem Bussche. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 10(3152676):10–5555, 2017.

[69] Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. *arXiv preprint arXiv:1905.09418*, 2019.

[70] Yufeng Wang, Shidong Zhou, Limin Xiao, Xiujun Zhang, and Jin Lian. Sparse bayesian learning based user detection and channel estimation for scma uplink systems. In *2015 International Conference on Wireless Communications & Signal Processing (WCSP)*, pages 1–5. IEEE, 2015.

[71] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

[72] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*, 2020.

[73] John I Yellott Jr. The relationship between luce's choice axiom, thurstone's theory of comparative judgment, and the double exponential distribution. *Journal of Mathematical Psychology*, 15(2):109–144, 1977.

[74] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *NIPS*, 2014.