HYPER-PARAMETER LEARNING FOR SPARSE STRUCTURED PROBABILISTIC MODELS

Tatiana Shpakova *Francis Bach* *Mike Davies*

INRIA - ENS INRIA - ENS University of Edinburgh

ABSTRACT

In this paper, we consider the estimation of hyperparameters for regularization terms commonly used for obtaining structured sparse parameters in signal estimation problems, such as signal denoising. By considering the convex regularization terms as negative log-densities, we propose approximate maximum likelihood estimation for estimating parameters for *continuous* log-supermodular distributions, which is a key property that many sparse priors have. We then show how "perturb-and-MAP" ideas based on the Gumbel distribution and efficient discretization can be used to approximate the log-partition function for these models, which is a crucial step for approximate maximum likelihood estimation. We illustrate our estimation procedure on a set of experiments with flow-based priors and signal denoising.

***Index Terms*—** hyperparameter learning, maximum likelihood, sparsity-inducing norms

1. INTRODUCTION

Structured sparsity has emerged a versatile tool to go beyond plain parsimonious models. Indeed, taking into account the potential structure between the signal coefficients to be set to zero, both interpretability and predictive performance can be improved. Structured sparse priors can be handled in the various frameworks that have emerged for sparse methods, within convex optimization [1, 2] or non-convex optimization [3, 4, 5].

While encoding structure has several benefits, it comes with the extra task of specifying a certain number of hyper-parameters, for example, for tree-structured sparsity, the weights to be given to each depth of the tree. The goal of this paper is to propose data-driven estimation procedures for all of these hyper-parameters for a class of priors based on submodular functions. These include tree-structured priors or group-based priors and are commonly used in signal estimation problems [6, 7].

In this paper, we make the following *contributions*:

– We propose in Section 2 approximate maximum likelihood estimation for estimating parameters in *continuous* log-supermodular distributions, whose negative log-densities are commonly used as structured sparse priors in signal processing applications.

– We show in Section 3 how "perturb-and-MAP" ideas based on the Gumbel distribution and efficient discretization can be used to approximate the log-partition function for these models, which is the key step for approximate maximum likelihood estimation. Here, the fact that submodular functions can be efficiently minimized is crucial.

– We illustrate our estimation procedure in Section 4 on a set of experiments with flow-based priors and signal denoising.

2. LOG-SUPERMODULAR DISTRIBUTIONS

As a probabilistic model, we are going to work with the family of log-supermodular distributions [8]. These distributions are a special case of a Gibbs distribution over some variable $x \in \mathcal{X}$, which could at this point be discrete or continuously valued:

$$dp(x) = \frac{e^{-f(x)}}{Z(f)} d\mu(x),$$

where $d\mu(x)$ is a base measure, and $f(x)$ is a potential function and the normalizer $Z(f) = \int_{\mathcal{X}} e^{-f(x)} d\mu(x)$. It is worth noting that the partition function $Z(f)$ is intractable in the general case, continuous or discrete, and its handling constitutes the core computational difficulty of probabilistic inference [9]. If the potential function $f(x)$ is submodular (see definition in the next section), then the distribution above is called log-supermodular (because $-f$ is supermodular).

We use these models as they cover a broad family of distributions and allow us to perform an efficient gradient descent optimization of the likelihood objective due to submodularity (see below).

2.1. Supermodular and Submodular Functions

Submodular functions can be defined on sets $\mathcal{X}$ which are products of intervals. These functions have the particular property to have polynomial-time minimization algorithms [10]. In this paper, we consider the special cases $\mathcal{X} = \mathbb{R}^n$ and $\mathcal{X} = \{0, 1\}^n$, which will lead respectively to continuous and discrete submodular functions.

Discrete functions. A function $f : \{0, 1\}^n \to \mathbb{R}$ can be uniquely identified to a *set-function* defined over subsets of

We acknowledge support the European Union's H2020 Framework Programme (H2020-MSCA-ITN-2014) under grant agreement n°642685 MacSeNet.

$\{1,\dots,n\}$, by defining $F(A) = f(1_A)$ where $1_A \in \{0,1\}^n$ is the indicator vector of the set A. It is then said submodular if it satisfies the following diminishing return property $\forall A, B \subseteq \{1,\dots,n\}$ such that $A \subseteq B$ and for all i, then $F(A \cup i) - F(A) \geq F(B \cup i) - F(B)$. Classical submodular functions include network flows, graph cuts, as well as concave functions of the cardinality and appear in many areas of signal processing and machine learning [11, 12]. In particular, non-decreasing submodular functions are commonly used to penalize the support of signal in compressed sensing [13]. Classical examples include group-based priors (counting the number of active groups) or tree-based priors (cardinality of the smallest rooted tree containing a given set), which are commonly used in signal processing [3].

Continuous functions. In the continuous setting, a twice differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ is submodular if for all $x \in \mathbb{R}^n$, the cross second-order derivatives are non-positive, that is, for all $i \neq j$, then $\frac{\partial^2 f}{\partial x_i \partial x_j}(x) \leqslant 0$. For non-differentiable functions, see more details in [14]. Important examples include the Lovász extensions of discrete submodular functions defined on $\{0,1\}^n$ [12, 11]. These extended functions are typically used for regularization as follows: when a vector w whose support $A = \{i, w_i \neq 0\}$ should have a small value $F(A)$ where F is the corresponding set-function, then the penalty $f(|w|)$, where f is the Lovász extension, and the absolute values are taken component-wise, is a natural convex relaxation [12, 13]. Classical examples are $f(|w|) = \sum_{A \in \mathcal{A}} d_A \max_{a \in A} |w_a|$, where $\mathcal{A}$ is a set of subsets of $\{1,\dots,n\}$ [15], see more details in Section 4.2. Other non-convex but tighter relaxations are of the form $f(|w|^p)$ where the power is taken component-wise [16].

2.2. Discrete Log-supermodular Distributions

An important example of a submodular minimization problem is a graphcut problem, which is very applicable in a variety of fields. For example it is popular to incorporate graphcut functions as potential functions in Markov random fields (MRFs) and perform an image segmentation as a side effect [17, 18]). Other examples can be mutual entropy, certain functions of eigenvalues of submatrices and many others [11]. Seeing them as log-densities thus leads to a probabilistic treatment (see, e.g., [8]), which similar to what we try to achieve in this paper for continuous distributions.

2.3. Continuous Log-supermodular Distributions

Several examples have been considered in other settings and then called "multivariate-totally positive of order 2", and include the multivariate logistic, Gamma distributions, as well as characteristic roots of random Wishart matrices [19]. We propose to extend their use in signal processing, in order to learn parameters of the associated negative log-densities (which are submodular functions), in structured sparsity problems.

2.4. Log-partition Function for Bayesian Learning

Let us consider the following standard denoising problem, with $x = D\alpha + \varepsilon$, where ε is a Gaussian noise $\sim \mathcal{N}(0, \sigma^2 I)$. Given the noisy signal $x \in \mathbb{R}^n$ and a dictionary $D \in \mathbb{R}^{n \times k}$, we try to recover the initial representation $\alpha \in \mathbb{R}^k$. The decoding problem often has the following form:

$$\min_{\alpha} \frac{1}{2}\|x - D\alpha\|^2 + \Lambda(\alpha),$$

where $\Lambda(\alpha)$ serves as a regularizer. This formulation can be considered from **three** points of view: 1) as penalized least squares regression without any probabilistic meaning, as 2) *maximum a posteriori* (MAP) or 3) *minimum-mean-square-error* (MMSE) estimation within some probabilistic model [20]. The probabilistic interpretation allows to learn parameters of $\Lambda(\alpha)$, which we aim to do here. The joint probability has the form: $P(x, \alpha) = P(\alpha)P(x|\alpha) = P(\alpha)P(\varepsilon = x - D\alpha)$, where $P(\alpha) = \frac{\exp(-\Lambda(\alpha))}{Z}$ is a log-supermodular prior on α. Thus, we would like to parameterize and learn the prior $P(\alpha)$ from the training dataset before performing the denoising on the test dataset.

In a sparse set-up we would like to encourage the argument α to contain plenty of zero elements and thus we can use the usual ℓ_1-formulation [21]:

$$\min_{\alpha} \frac{1}{2}\|x - D\alpha\|^2 + \Lambda(\alpha),$$
$$s.t.\ \Lambda(\alpha) = f(|\alpha|),$$

where $|\alpha|$ is meant component-wise. Here $f(|\alpha|)$ is encoding the structured sparsity, see an example in the Section 4.2. In this paper, for simplicity, we consider only orthonormal dictionaries.

Maximum likelihood for parameter learning. The main goal here is to consider $\Lambda(\alpha)$ as a negative log-density with the functional parameter $f(\cdot)$ and to find its optimal data-dependent parameters by maximum likelihood. To do so, we need to solve an optimization problem that involves the log-partition function in the way described below. We use the notation, for a submodular function f defined on $\mathbb{R}^k_+$, $\log Z(f) = A(f) = \log \int_{\mathbb{R}^k_+} \exp(-f(\alpha))d\alpha$. Note that, by a simple change of variable, we have:

$$\log \int_{\mathbb{R}^k} \exp(-f(|\alpha|))d\alpha = k \log 2 + \log \int_{\mathbb{R}^k_+} \exp(-f(\alpha))d\alpha.$$

We thus have $-\log P(\alpha) = f(\alpha) + A(f) + k \log 2$.

In order to learn the distribution of α we need to perform maximum likelihood for the densities defined above. To perform this we need to approximate the log-partition function A, which is always convex, but usually hard to compute. Two questions arise: (1) approximation of $A(f)$, and (2) parameterization of f in a suitable form so that our optimization problems are easily solved.

3. PERTURB-AND-MAP

To perform an effective parameter learning, we would like to approximate the log-partition function $A(f)$ by making use of the "perturb-and-MAP" approach [22]. One necessary point is to have an access to an efficient MAP-oracle, e.g., a graphcut solver for our particular example of flow-based priors (see Section 4.2).

Our algorithm is based on an approximation result from [22], which states that for any real-valued function g defined on a discrete set $\mathcal{T} = \prod_{i=1}^{n} \mathcal{T}_i$, then

$$\log \sum_{t \in \mathcal{T}} e^{g(t)} \leqslant \mathbb{E}_{h_1,\dots,h_n \sim \mathrm{Gumbel}} \Big[\max_{t \in \mathcal{T}} \Big(g(t) + \sum_{i=1}^{n} h_i(t_i) \Big) \Big],$$

where Gumbel denotes the Gumbel distribution[1] and with a collection $\{h_i(t_i)\}_{t_i \in \mathcal{T}_i}^{i=1,\dots,n}$ of independent Gumbel samples. This allows to define an upper-bound on the log-partition function, which is based on perturbing g and performing maximization; it is thus efficient only for functions g for which adding a separable terms leads to efficient optimization. This is exactly the case for negatives of submodular functions.

Via the proposed approximation we achieve a direct way of parameter learning in the discrete case. We can approximate the expectation over Gumbels using Monte-Carlo ideas by replacing the expectation by sampling. An efficient number of Gumbel samples depends on application, however in our experiments, $M = 100$ seems to be enough for most of the setups. Following, [23], in practice, when embedding the approximation result above in an optimization problem (for maximum likelihood) we can use stochastic gradient descent as a subroutine [23], rather than using a fixed set of Gumbel samples.

3.1. Extension to Continuous Case

To work with continuous data, we need to perform a discretization of the partition function to cast its calculation as a discrete optimization problem.

Approximation of $A(f) = \log \int_{\mathbb{R}_+^k} \exp(-f(\alpha)) d\alpha$. In order to approximate $A(f)$, we are going to discretize each α_i into r values $0 = u_0 < u_1 < \cdots < u_{r-1}$, and consider the measure on $\mathbb{R}_+$ define as the weighted sum of Diracs $\sum_{j=0}^{r-1} \pi_j \delta(\beta = u_j)$. A simple example (based on the trapezoidal rule) for the weights (π_j) is $\pi_0 = \frac{u_1 - u_0}{2}$, $\pi_{r-1} = \frac{u_{r-1} - u_{r-2}}{2}$, with the rest as $\pi_j = \frac{u_{j+1} - u_{j-1}}{2}$. We then discretize the integral in the following way

$$\hat{A}(f) = \log \sum_{z \in \{0,\dots,r-1\}^k} \Big(\prod_{i=1}^{k} \pi_{z_i} \Big) \exp(-f(u_{z_i}, \dots, u_{z_k})).$$

[1] The Gumbel distribution on the real line has the cumulative distribution function $F(z) = \exp(-\exp(-(z + c)))$, where c is the Euler constant.

This is done by considering kr Gumbel variables $h_{i,j}$, $i \in \{1, \dots, k\}$ and $j \in \{0, \dots, r-1\}$, and the perturb-and-MAP approximation

$$\hat{A}_G(f) = \mathbb{E}_h \Big(\max_z \Big\{ -f(\cdot) + \sum_{i=1}^{k} h_{i,z_i} + \sum_{i=1}^{k} \log \pi_{z_i} \Big\} \Big).$$

The separable term $h_{i,z_i} + \log \pi_{z_i}$ does not change the nature of the problem and the function remains submodular (but now defined on a finite number of values).

3.2. Decoding with MMSE

In this section we discuss several ways of performing inference. Decoding with MAP is straightforward, as we can get the approximation with a discrete solver or an exact solution via divide-and-conquer algorithms [24], if the objective is convex. Instead of using MAP decoder we can use MMSE, which is known to be the Bayesian optimal classifier for the ℓ_2-loss function (see, e.g., [20]). However, it is a challenge to calculate, and as one of our contributions in this work we propose a way of its approximation via perturb-and-MAP ideas. Indeed, we have the estimator

$$\psi_{MMSE}(x) = \mathbb{E}(\alpha|x)$$
$$P(\alpha|x) = \frac{P(x,\alpha)}{\sum_\alpha P(x,\alpha)} = \frac{\exp(-s(\alpha,x))}{Z(s)},$$

where $s(\alpha, x) = \frac{1}{2\sigma^2}||x - D\alpha||^2 + \Lambda(\alpha) = \frac{1}{2\sigma^2}||D^T x - \alpha||^2 + \Lambda(\alpha)$ for an orthonormal dictionary D. Then, the optimal estimator can be computed as:

$$\psi_{MMSE}(x) = \mathbb{E}(\alpha|x) \approx$$
$$\frac{1}{M} \sum_{m=1}^{M} \arg\min_{\alpha_m} \Big\{ s(\alpha_m) - \sum_k h_k(\alpha_{km}) \Big\},$$

as the expectation of the sufficient statistics is known to be a gradient of the log-partition function [25]. Thus, we can get an approximate solution via approximate mean marginals and discretization.

4. EXPERIMENTS

In the experiment section we illustrate the proposed parameter learning technique and compare the performance of two decoding approaches.

4.1. Synthetic Data. Experiments on decoding

We consider a one dimensional distribution $P(\alpha) = \frac{\exp^{-|\alpha|}}{Z}$. Closed-form solution for MMSE in discrete setup can then be calculated exactly for this one dimensional setup. We present denoising results for 10,000 randomly sampled data points in Table 1. We can clearly see that MMSE outperforms MAP in this synthetic setup, in a set-up where the Gumbel approximation of the log-partition function is exact.

Approach	ℓ_2 loss value
continuous MAP solution	0.693
MAP-oracle	0.744
MMSE exact	0.626
MMSE approx [$M = 100$]	0.632
MMSE approx [$M = 1000$]	0.628

Table 1: Denoising results. Synthetic data.

4.2. Flow-based priors

We now show examples where our maximization problems can be cast a a max-flow / min-cut. Following Sections 6.3 and 6.4 of [12], we consider prior $\Lambda(\alpha)$ as functions f of the form

$$f(\alpha) = \sum_{A \in \mathcal{A}} d_A \max_{a \in A} |\alpha_a|,$$

where d_A are parameters that we are interested of. In order to minimize the function $f(u_{z_1}, \dots, u_{z_k}) + \sum_{i=1}^{k} v_{i,z_i}$, which is required in Section 3.1, following [26], we can create a weighted directed graph where the (st)-minimum cut gives the optimal solution, thus making the optimization efficient.

4.3. Real Data. Experiments on the parameter learning and decoding.

We work with patches 8×8 of the 'boat' image. Half of them are considered as a train dataset to learn hyperparameteters of our priors, and another half as a test dataset. The original image is 512×512. After Gaussian noise was added to the test dataset, we try to denoise it with the proposed decoding techniques. We compare two ways of denoising: 1) model the prior directly for the pixels values and 2) model the prior for some wavelet coefficients.

We work under the following assumption $P(\alpha) = \frac{e^{-f(\alpha)}}{Z(f)}$, where $f(\alpha) = \sum_{A \in \mathcal{A}} d_A \max_{a \in A} |\alpha_a|$, where $\mathcal{A}$ is a set of predefined groups of variables and α could be either the pixels values, either the wavelet coefficients. These groups could be based on the image grid or on the wavelet tree structure. Moreover, D is a orthogonal wavelet transform (we use Haar wavelets).

Before we report results in Tables 2 and 3, we first comment on the discretization grid which influences a lot the performance. We use a uniform grid with a step Δ (either 1 or 10). The quality and the running time consumption of the discrete MAP and MMSE depends on the discretization grid and on Δ correspondingly. For a fair comparison we spend the same amount of time for both: MAP and $\Delta = 1$ consume approximately the same amount of time as MMSE and $\Delta = 10$. For MMSE we use $M = 100$ Gumbel perturbations. The method "c. MAP" refers to the exact MAP solution achieved through direct continuous optimization via divide-and-conquer algorithm [24]. In Tables 2 and 3 we compute signal-to-noise ratios (SNR), where SNR $= 20 \log_{10} \frac{||x||_2}{||\hat{x}-x||_2}$ and x is a test image and $\hat{x}$ its prediction. We accompany the results with their standard deviation, across 10 different noise samples.

First set of experiments. We compare several models: "baseline", "unary" and "grid" models. "baseline" is a model with no learning and we set parameters d_A equal to zeros, "unary" corresponds to groups of size one (independent but not identically distributed variables), and "grid" corresponds to groups of size one and two in a grid manner.

method	baseline	unary	grid
c. MAP	$\mathbf{22.33} \pm 0.02$	$\mathbf{22.32} \pm 0.02$	$\mathbf{26.16} \pm 0.02$
$\text{MAP}_{\Delta 10}$	21.95 ± 0.01	21.95 ± 0.01	25.36 ± 0.02
$\text{MAP}_{\Delta 1}$	$\mathbf{22.32} \pm 0.02$	$\mathbf{22.31} \pm 0.02$	$\mathbf{26.16} \pm 0.02$
MMSE	$\mathbf{22.31} \pm 0.02$	$\mathbf{22.30} \pm 0.01$	25.99 ± 0.01
neg. lhood	356.4	355.7	253.5

Table 2: Primal approach. Signal-to-noise ratio values.

Second set of experiments. The model "tree" corresponds to groups of size one and two that represent the wavelet quad-tree dependencies of two-dimensional Haar wavelets.

method	baseline	unary	grid
c. MAP	$\mathbf{22.33} \pm 0.02$	24.96 ± 0.01	24.66 ± 0.01
$\text{MAP}_{\Delta 10}$	21.98 ± 0.01	24.63 ± 0.08	24.44 ± 0.05
$\text{MAP}_{\Delta 1}$	$\mathbf{22.33} \pm 0.02$	25.02 ± 0.03	24.72 ± 0.03
MMSE	$\mathbf{22.29} \pm 0.03$	$\mathbf{25.17} \pm 0.02$	$\mathbf{25.15} \pm 0.02$
neg. lhood	480	185.5	183.0

Table 3: Wavelets approach. Signal-to-noise ratio values.

Summary: From Tables 2 and 3, we can see that we are able to learn the structure. In the primal approach (directly on pixels), the Markov random field ("grid") performs best, while for the wavelet approach, unary potentials already work best (with little gains in likelihood and denoising performance for the "tree" approach). Note that MMSE sometimes outperform MAP, however it depends on the discretization grid.

5. CONCLUSION

We proposed a new parameter learning approach in the non-trivial structured and continuous setup. We demonstrated its performance for a denoising problem. We also propose a way to perform approximate MMSE estimation which is the optimal decoder for ℓ_2-loss evaluation. There is still some space for further investigation: non-uniform discretization grids, ℓ_p-looking norms such as done by [27], where more heavy-tailed priors were considered.

6. REFERENCES

[1] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 1, pp. 49–67, 2006.

[2] L. Jacob, G. Obozinski, and J.-P. Vert, "Group lasso with overlap and graph lasso," in *Proceedings of the 26th annual international conference on machine learning*. ACM, 2009, pp. 433–440.

[3] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde, "Model-based compressive sensing," *IEEE Transactions on Information Theory*, vol. 56, no. 4, pp. 1982–2001, 2010.

[4] L. He and L. Carin, "Exploiting structure in wavelet-based bayesian compressive sensing," *IEEE Transactions on Signal Processing*, vol. 57, no. 9, pp. 3488–3497, 2009.

[5] J. Huang, T. Zhang, and D. Metaxas, "Learning with structured sparsity," *Journal of Machine Learning Research*, vol. 12, no. Nov, pp. 3371–3412, 2011.

[6] S. Kim and E. P. Xing, "Tree-guided group lasso for multi-task regression with structured sparsity.," in *ICML*, 2010, pp. 543–550.

[7] P. Zhao and B. Rocha, G.and Yu, "Grouped and hierarchical model selection through composite absolute penalties," *Department of Statistics, UC Berkeley, Tech. Rep*, vol. 703, 2006.

[8] J. Djolonga and A. Krause, "From MAP to Marginals: Variational Inference in Bayesian Submodular Models," in *Adv. NIPS*, 2014.

[9] M. J. Wainwright, M. I. Jordan, et al., "Graphical models, exponential families, and variational inference," *Foundations and Trends® in Machine Learning*, vol. 1, no. 1–2, pp. 1–305, 2008.

[10] A. Krause and D. Golovin, "Submodular function maximization," in *Tractability: Practical Approaches to Hard Problems*. Cambridge University Press, February 2014.

[11] S. Fujishige, *Submodular Functions and Optimization*, Annals of discrete mathematics. Elsevier, 2005.

[12] F. Bach, "Learning with submodular functions: a convex optimization perspective," *Foundations and Trends in Machine Learning*, vol. 6, no. 2-3, pp. 145 – 373, 2013.

[13] G. Obozinski and F. Bach, "Convex relaxation for combinatorial penalties," *arXiv preprint arXiv:1205.1240*, 2012.

[14] F. Bach, "Submodular functions: from discrete to continuous domains," *Mathematical Programming*, pp. 1–41, 2016.

[15] F. Bach, R. Jenatton, J. Mairal, G. Obozinski, et al., "Structured sparsity through convex optimization," *Statistical Science*, vol. 27, no. 4, pp. 450–468, 2012.

[16] M. El Halabi, F. Bach, and V. Cevher, "Combinatorial penalties: Which structures are preserved by convex relaxations?," in *International Conference on Artificial Intelligence and Statistics*, 2018, pp. 1551–1560.

[17] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts?," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 26, no. 2, pp. 147–159, 2004.

[18] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 26, no. 9, pp. 1124–1137, 2004.

[19] S. Karlin and Y. Rinott, "Classes of orderings of measures and related correlation inequalities. i. multivariate totally positive distributions," *Journal of Multivariate Analysis*, vol. 10, no. 4, pp. 467–498, 1980.

[20] R. Gribonval, "Should penalized least squares regression be interpreted as maximum a posteriori estimation?," *IEEE Transactions on Signal Processing*, vol. 59, no. 5, pp. 2405–2410, 2011.

[21] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Review*, vol. 43, no. 1, pp. 129–159, 2001.

[22] T. Hazan and T. Jaakkola, "On the partition function and random maximum a-posteriori perturbations," in *Proc. ICML*, 2012.

[23] T. Shpakova, F. Bach, and A. Osokin, "Marginal weighted maximum log-likelihood for efficient learning of perturb-and-map models," .

[24] H. Groenevelt, "Two algorithms for maximizing a separable concave function over a polymatroid feasible region," *European Journal of Operational Research*, vol. 54, no. 2, pp. 227–236, 1991.

[25] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," *Foundations and Trends in Machine Learning*, vol. 1, no. 1-2, pp. 1–305, 2008.

[26] H. Ishikawa, "Exact optimization for markov random fields with convex priors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, pp. 1333–1336, 2003.

[27] N. Shervashidze and F. Bach, "Learning the structure for structured sparsity," *IEEE Transactions on Signal Processing*, vol. 63, no. 18, pp. 4894–4902, 2015.