Sparse Markov Source Estimation via Transformed Lasso

Teemu Roos
Helsinki Institute for Information Technology HIIT
Univ. of Helsinki & Helsinki Univ. of Technology, Finland
Email: teemu.roos@hiit.fi

Bin Yu
Department of Statistics
University of California, Berkeley, USA
Email: binyu@stat.berkeley.edu

***Abstract*—We establish a connection between Lasso-type ℓ_1 regularization and learning variable length Markov chains (VLMCs). This is achieved by a parameterization of discrete-valued finite-memory Markov sources in which setting a parameter value equal to zero is equivalent to eliminating a node in the corresponding context tree model. The parameterization involves a Haar wavelet transformation on a set of indicator functions, the output of which is mapped to symbol probabilities via logistic regression. The optimization problem is convex and can be solved efficiently using existing tools. We present preliminary results, comparing the method to an earlier algorithm for learning VLMCs in terms of model selection and prediction performance. We also discuss other transformations which lead to a flexible family of sparse representations of Markov sources.**

I. Introduction

The model class of variable length Markov chains (VLMC), or context tree models, is extensively studied in information theory, see e.g. [1]–[4]. We propose a parameterization of discrete-valued finite-memory Markov sources in which VLMC models arise naturally as a result of regularization of the ℓ_1 norm of the parameters—as in Basis Pursuit and Lasso [5], [6] for which efficient path following algorithms exist [7], [8]. The parameterization involves a Haar wavelet transformation on a set of indicator functions, the result of which is mapped to symbol probabilities via logistic regression. The resulting ℓ_1-penalized logistic regression is convex and there are algorithms that can be used to solve it efficiently [9], [10].

II. Context Tree Models

Let $\mathbf{X} = (X_1, \ldots, X_n)$ be a discrete random sequence of length n, where each symbol (variable) X_i, $i \leq n$ takes value from a fixed alphabet $\mathcal{X}$. We denote the subsequence of $\mathbf{X}$ starting at time i and ending at time j by $\mathbf{X}_i^j$.

By the chain rule, we can write the probability of a sequence as a product of individual symbols:

$$P(\mathbf{X} = \mathbf{x}) = \prod_{i=i}^{n} P(X_i = x_i \mid \mathbf{X}_1^{i-1} = \mathbf{x}_1^{i-1}).$$

A kth order Markov model assumption reduces the relevant part of the history in the conditioning part, or the *context*, from $\mathbf{X}_1^{i-1}$ to $\mathbf{X}_{i-k}^{i-1}$:

$$P(\mathbf{X} = \mathbf{x}) = \prod_{i=i}^{n} P(X_i = x_i \mid \mathbf{X}_{i-k}^{i-1} = \mathbf{x}_{i-k}^{i-1}).$$

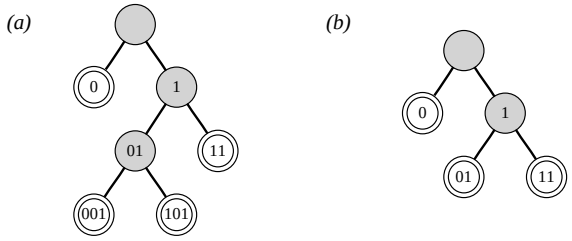

Fig. 1. Examples of context trees. In each tree, left and right branches correspond to zero and one, respectively. The shaded nodes indicate the relevant symbols, and the resulting contexts are given in the terminal nodes. If $X_{i-1} = 0$, then the context is simply 0 in both trees. On the other hand, if $X_{i-1} = 1$ and $X_{i-2} = 0$, then the context is 01 in tree *(b)*, but one more bit is needed decide in tree *(a)*.

The initial part with $i < k+1$ can be treated, for instance, by assuming a fixed initial sequence, $\mathbf{X}_{-\infty}^0$, prefixed to the actual sequence, so that the context $\mathbf{X}_{i-k}^{i-1}$ is always well-defined. For fully parameterized discrete models, the number of parameters needed to specify the model is $|\mathcal{X}|^k(|\mathcal{X}| - 1)$. This makes it hard to estimate the parameters accurately for $k \gg 1$.

In variable length Markov chains (VLMC), the number of relevant preceding symbols depends on the context. VLMC models can be characterized by *context trees*, examples of which are shown in Fig. 1. Given a context tree, the number of parameters needed is $L(|\mathcal{X}| - 1)$, where L is the number of leaf nodes in the tree. Ordinary Markov models are a special case of VLMCs where the context tree is a balanced tree of depth k with $L = |\mathcal{X}|^k$ leaf nodes.

It is convenient, for the present purpose, to use a logistic (regression) parameterization so that a distribution over $\mathcal{X} = \{0, 1\}$ is specified by the formula

$$P(X_i = 1 \mid \mathbf{X}_{i-k}^{i-1} = \mathbf{x}_{i-k}^{i-1}) = \frac{1}{1 + e^{-\boldsymbol{\beta}' \mathbf{z}(\mathbf{x}_{i-k}^{i-1})}}, \qquad (1)$$

where $\boldsymbol{\beta} \in \mathbb{R}^d$ is a parameter vector, and $\mathbf{z}(\mathbf{x}_{i-k}^{i-1}) \in \mathbb{R}^d$ is a vector of coefficients determined by the context (to be defined more precisely below, see Eq. (2)). The parameterization is complete in the sense that any conditional distribution can be represented by choosing suitable parameter values. It is well known that logistic regression models are exponential families, which implies that the log-likelihood function is concave, see e.g. [11].

For the sake of simplicity, we concentrate on the binary

case. Generalization to larger alphabets, i.e., multiple logistic regression, can be achieved by the group Lasso [12], using techniques from [10].

Consider $\mathbf{z}$ formed by concatenating indicators for each possible vector $\mathbf{x}_{i-k}^{i-1}$. In this way, for order $k = 3$, the $2^3 = 8$ different coefficients vectors (sorted according to the "least significant bits"; the reason for this will become clear) are given by

$$\begin{array}{cc} \mathbf{x}_{i-k}^{i-1} & \mathbf{z}(\mathbf{x}_{i-k}^{i-1}) \\ \hline 000 & (1,0,0,0,0,0,0,0) \\ 100 & (0,1,0,0,0,0,0,0) \\ 010 & (0,0,1,0,0,0,0,0) \\ \vdots & \vdots \\ 111 & (0,0,0,0,0,0,0,1) \end{array} \tag{2}$$

With this mapping, we can represent any conditional distribution $P(X_i \mid \mathbf{X}_{i-k}^{i-1} = \mathbf{x}_{i-k}^{i-1})$ by defining parameters

$$\beta(\mathbf{x}_{i-k}^{i-1}) = \ln \frac{P(X_i = 1 \mid \mathbf{X}_{i-k}^{i-1} = \mathbf{x}_{i-k}^{i-1})}{P(X_i = 0 \mid \mathbf{X}_{i-k}^{i-1} = \mathbf{x}_{i-k}^{i-1})}$$

and concatenating these parameters into a single vector $\boldsymbol{\beta}$ in the same order as the contexts are listed in the above table, i.e., $\boldsymbol{\beta}' = (\beta(000),\ \beta(100),\ \beta(010),\ \ldots,\ \beta(111))$. The dot product $\boldsymbol{\beta}'\mathbf{z}$ simply picks the correct parameter from vector $\boldsymbol{\beta}$, and Eq. (1) yields the desired probability.

III. Transformed Lasso

In the Lasso (least absolute shrinkage and smoothing operator) [6], the log-likelihood is penalized by the ℓ_1 norm of the parameters[1]:

$$\max_{\boldsymbol{\beta}} \log P(\mathbf{x}\ ;\ \boldsymbol{\beta}) - \lambda \|\boldsymbol{\beta}\|_1. \tag{3}$$

In the standard linear-quadratic case, the log-likelihood is a quadratic function of the parameters. The ℓ_1 penalty has the property that usually many of the parameter estimates are equal to zero. Assuming that the log-likelihood function is downwards convex, as it is in the logistic regression case, the optimization problem can be solved efficiently by convex optimization methods; for the linear–quadratic case, see [6]–[8].

Under the logistic parameterization (1), letting parameter $\beta(\mathbf{x}_{i-k}^{i-1})$ be (close to) zero, results in a (nearly) uniform conditional distribution for X_i given $\mathbf{X}_{i-k}^{i-1} = \mathbf{x}_{i-k}^{i-1}$. Thus, ℓ_1 penalization tends to *smooth* the parameter estimates towards the uniform distribution. However, this is not the only kind of sparsity (or "simplicity") we usually expect from Markov models: we would also like the *differences* between parameters to be small. For instance, if $\beta(001) = \beta(101)$, then X_{i-3} has no effect given that $\mathbf{X}_{i-2}^{i-1} = 01$. In Fig. 1, enforcing the equality of these two parameters changes tree *(a)* into tree *(b)*.

In the *fused Lasso* [14], one also penalizes for the absolute difference of subsequent parameters (ordered in a suitable fashion):

$$\max_{\boldsymbol{\beta}} \log P(\mathbf{x}\ ;\ \boldsymbol{\beta}) - \lambda_1 \|\boldsymbol{\beta}\|_1 - \lambda_2 \sum_{j=2}^{p} |\beta_j - \beta_{j-1}|, \tag{4}$$

where p is the number of parameters, and λ_1 and λ_2 are regularization parameters. If the parameters are ordered as above (see Eq. (2)), this would indeed favor to some extent parameter vectors where subsequent contexts give identical conditional distributions. This handles the case where the identity $\beta(000) = \beta(100)$ eliminates the effect of X_{i-3} in a given context. However, it is unclear what the meaning of, say, $\beta(110) = \beta(001)$ is: the contexts 110 and 001 are at first sight the opposite of each other; why should we have a bias that encourages their giving the same distribution on X_i? On the other hand, penalizing only the pairs that differ in X_{i-k} but agree in the other symbols[2], will penalize *only* the effect X_{i-k}.

More generally, we should of course penalize for the absolute difference between any two parameters which we like to be (almost) equal. Adding very many penalty terms will, however, slow down the optimization procedure. Our approach, which we outline next, is based on penalization of linear combinations of the logistic parameters and avoids the explicit use of additional pair-wise and higher-order penalties.

What we propose to do is to perform a suitable linear transformation on the parameters and use Lasso (ℓ_1) penalization on the *transformed* parameters. This yields the optimization problem:

$$\max_{\boldsymbol{\beta}} \log P(\mathbf{x}\ ;\ \boldsymbol{\beta}) - \lambda \|T\boldsymbol{\beta}\|_1, \tag{5}$$

where $\lambda > 0$ is a regularization parameter, and T is a linear transformation. We call this method the *transformed Lasso*.

The idea is that if the (original) parameter vector $\boldsymbol{\beta}$ has some smoothness properties captured by the transformation T, then $T\boldsymbol{\beta}$ is sparse, i.e., it has only a few non-zero coefficients. When estimating the parameters from data using (5), the estimates of these parameters tend to be set to zero. Since T is a linear transformation, the concavity of the penalized log-likelihood is retained.

In fact the transformed Lasso was proposed already by Tibshirani *et al.* [14] (using the Haar transformation) for the linear–quadratic case, but the authors found it ill-suited, mainly because in their example, the predictor structure was not 'dyadic', like it is in our case: while the predictors were ordered so that they formed blocks, i.e., consecutive runs of identical coefficients, the block boundaries did not occur near powers of two. This implies that the parameter vector is not necessarily sparse in the Haar domain. Due to the way we order the parameters in Markov models, we get such a dyadic structure whenever the model allows a context-tree

[1] Alternatively, we can maximize the log-likelihood subject to an upper-bound on the ℓ_1 norm of $\boldsymbol{\beta}$. A review of Lasso and related methods is given in [13].

[2] For $k = 3$, this means that we penalize by the sum $|\beta(000) - \beta(100)| + |\beta(010) - \beta(110)| + |\beta(001) - \beta(101)| + |\beta(011) - \beta(111)|$; compare to the last term of the fused Lasso, Eq. (4).

$$\mathcal{H} = \begin{bmatrix} 1/\sqrt{8} & \times & (1, & 1, & 1, & 1, & 1, & 1, & 1, & 1) \\ 1/\sqrt{8} & \times & (1, & 1, & 1, & 1, & -1, & -1, & -1, & -1) \\ 1/2 & \times & (1, & 1, & -1, & -1, & 0, & 0, & 0, & 0) \\ 1/2 & \times & (0, & 0, & 0, & 0, & 1, & 1, & -1, & -1) \\ 1/\sqrt{2} & \times & (1, & -1, & 0, & 0, & 0, & 0, & 0, & 0) \\ 1/\sqrt{2} & \times & (0, & 0, & 1, & -1, & 0, & 0, & 0, & 0) \\ 1/\sqrt{2} & \times & (0, & 0, & 0, & 0, & 1, & -1, & 0, & 0) \\ 1/\sqrt{2} & \times & (0, & 0, & 0, & 0, & 0, & 0, & 1, & -1) \end{bmatrix} \tag{6}$$

representation with less than the full number of nodes. This is illustrated in an example below.

IV. Haar transformation

For details of the Haar and other wavelet transformations, see e.g. [15]. The Haar transform matrix of order 8 (to be used for models with 8 parameters), which gives the *basis vectors* as its rows, is given by Eq. (6). The multipliers on the left make sure that each vector $\mathbf{u}_i$ is of unit length. The basis vectors are also orthogonal in the sense that $\mathbf{u}_i'\mathbf{u}_j = 0$ for all $i \neq j$. The (*forward*) Haar transform is mathematically equivalent to multiplying a vector by the Haar matrix, $\mathbf{v} \mapsto \mathcal{H}\mathbf{v}$, whereas the *inverse* Haar transform is equivalent to multiplying by the inverse of the matrix, which is by orthogonality equivalent to the transpose, $\mathbf{w} \mapsto \mathcal{H}'\mathbf{w}$.

In practice, matrix multiplications are not used, since a simple and fast algorithm exist for the Haar transform and its inverse. The computational complexity of the fast Haar transform is linear in the length of the transformed sequence, i.e., the number of coefficients in the model.

The linear transform T in (5 is then the Haar transform $\boldsymbol{\beta} \mapsto \mathcal{H}\boldsymbol{\beta}$, where $\mathcal{H}$ is the Haar matrix. This gives a representation of the parameter vector in terms of the Haar basis vectors. The first basis vector represents the mean of all the parameters, which gives the general *bias* towards $X_i = 1$. The second basis function gives the *difference* in bias towards $X_i = 1$ between the cases where $X_{i-1} = 0$ and those where $X_{i-1} = 1$. The third one gives the difference in bias between the cases where $\mathbf{X}_{i-2}^{i-1} = 00$ and those where $\mathbf{X}_{i-2}^{i-1} = 10$, and so on. In a given context, the bias is obtained as a sum of (possibly negated) coefficient values.

For instance, any Markovian model of order 2 can be represented using basis vectors 1,2,3, and 4 only. Any distribution compatible with the three-node context tree *(b)* in Fig. 1, where the different contexts are $\{0, 01, 11\}$, can be represented using three basis vectors, namely vectors 1, 2, and 4. In the binary case, the number of basis vectors needed to represent any VLMC is equal to L, the number of leaf nodes in the corresponding context tree, which is optimal.

On the other hand, since the number of context trees of maximum depth k is less than the number of subsets of k basis vectors, there are some subsets that correspond to no context tree. For instance, by setting certain coefficients to zero and retaining others, it is possible to eliminate the *individual* effect of X_{i-1} while retaining the effect of X_{i-2} given that $X_{i-1} = 0$, etc. In other words, it is possible for the effect of a symbol to cancel out *on the average* (when averaged over all contexts) even when symbols further away in the context have a non-zero effect. We defer further discussion about when this is useful and when not to the full version of this paper.

We now describe how the transformed Lasso problem (5) can be solved by existing Lasso techniques. Denoting the transformed parameters by $\boldsymbol{\eta} = \mathcal{H}\boldsymbol{\beta}$, the problem can be rewritten as

$$\max_{\boldsymbol{\eta}} \log P(\mathbf{x}\,;\, \mathcal{H}'\boldsymbol{\eta}) - \lambda\|\boldsymbol{\eta}\|_1, \tag{7}$$

where we used $\mathcal{H}'\boldsymbol{\eta} = \boldsymbol{\beta}$. Now, consider the likelihood function written in terms of the $\boldsymbol{\eta}$-parameters,

$$\begin{aligned} P(X_i = 1 \mid \mathbf{X}_{i-k}^{i-1} = \mathbf{x}_{i-k}^{i-1}) &= 1/\left(1 + e^{-(H'\boldsymbol{\eta})'\mathbf{z}(\mathbf{x}_{i-k}^{i-1})}\right) \\ &= 1/\left(1 + e^{-\boldsymbol{\eta}'\mathcal{H}\mathbf{z}(\mathbf{x}_{i-k}^{i-1})}\right). \end{aligned} \tag{8}$$

From (7) and (8) we see that the transformed problem is equivalent to the usual Lasso problem with the inputs given by $\mathcal{H}\mathbf{z}(\mathbf{x}_{i-k}^{i-1})$, i.e., the inputs are simply mapped through the Haar transform. Having solved the optimization problem, the optimal parameter vector $\hat{\boldsymbol{\beta}}_\lambda$ can be obtained by the inverse transform, $\hat{\boldsymbol{\beta}}_\lambda = \mathcal{H}'\hat{\boldsymbol{\eta}}_\lambda$, where $\hat{\boldsymbol{\eta}}_\lambda$ is the solution of (7).

In our application, it turns out that it is better to omit the scaling multipliers in (6). This is because the higher-order basis functions, like the four bottom rows in (6), are multiplied by a factor which is exponential (in the order of the effect) with respect to the factor of the lowest-level basis functions. Thus, a small change in a parameter associated with the higher-order effects alters the resulting probabilities much more than a change in a lower-order parameter. Unless this is accounted for in the ℓ_1 penalty term, the outcome is that many spurious high-order effects enter the model at the expense of some actual lower-order terms being omitted, which affects the performance severely. The problem is easily fixed by modifying the fast Haar transform so as to ignore the normalizing multipliers. We omit the details.

V. Simulation Experiment

We present some preliminary results illustrating the parameter transformation approach. Data was generated by sampling random binary sequences of given length from a fixed VLMC model described below. The maximum order of the effects in the transformed Lasso method was restricted to $k = 7$. We compare the estimated models to the generating model,

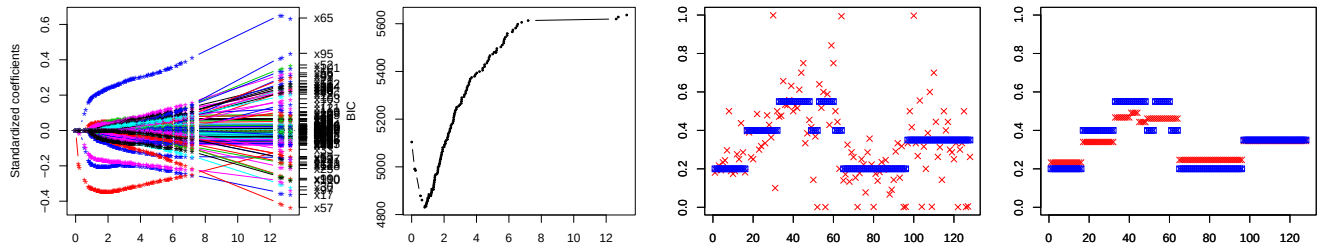

Fig. 2. *(a)* The regularization path obtained by `glmpath`. Penalized coefficients are plotted against optimization step. *(b)* The BIC criterion plotted against optimization step. *(c)* Maximum likelihood and *(d)* penalized estimates. The true parameter values, which are the same in both panels, are shown with blue squares (□), the estimates with red crosses (×). Data was generated from the model in Example 1, with sample size $n = 2048$.

and also estimate the negative log-likelihood by evaluating the per-symbol logarithmic prediction errors in a test sequence sampled from the same distribution.

For solving the transformed Lasso problem (5), we used the `glmpath` package[3] [9], which also gives a regularization path, i.e., the set of solutions obtained by letting the regularization parameter λ vary between some maximum value and zero. Having computed the regularization path, we selected the level of regularization, λ, by the Bayesian information criterion (BIC), see e.g. [16].

Example 1: Let the model be

$$P(X_i = 1 \mid \mathbf{X}_{i-k}^{i-1} = \mathbf{x}_{i-k}^{i-1}) = \begin{cases} 0.2 & \text{if } \mathbf{x}_{i-3}^{i-1} = 000 \\ 0.4 & \text{if } \mathbf{x}_{i-3}^{i-1} = 100 \\ 0.55 & \text{if } \mathbf{x}_{i-3}^{i-1} = 010 \\ 0.4 & \text{if } \mathbf{x}_{i-5}^{i-1} = 00110 \\ 0.55 & \text{if } \mathbf{x}_{i-5}^{i-1} = 10110 \\ 0.55 & \text{if } \mathbf{x}_{i-5}^{i-1} = 01110 \\ 0.4 & \text{if } \mathbf{x}_{i-5}^{i-1} = 11110 \\ 0.2 & \text{if } \mathbf{x}_{i-2}^{i-1} = 01 \\ 0.35 & \text{if } \mathbf{x}_{i-2}^{i-1} = 11 \end{cases}$$

Figure 2 shows the regularization path and the BIC curve in a representative run with sample size $n = 2048$, together with the obtained maximum likelihood and penalized parameter estimates. It can be seen that the maximum likelihood estimates in panel *(c)* (obtained by setting $\lambda = 0$) are very noisy; many of them are either zero or one since they are associated with contexts where only one of the outcomes has occurred. The transformed Lasso estimates, panel *(d)*, are much more stable and give a better estimate of the true parameters.

In order to assess the model selection and prediction (compression) performance of the transformed Lasso, we use an implementation of the Context algorithm [1] available as the `VLMC` package[4], see [3], as a baseline method. The BIC criterion was used also in this case for choosing the complexity of the model[5].

[3]R package available from CRAN, http://cran.R-project.org/.

[4]Available from CRAN.

[5]While BIC is generally better than the Akaike information criterion (AIC) in model selection, the reverse tends to hold for prediction. However, in our case BIC is better in both.

Figure 4 summarizes the models learned by the transformed Lasso method and VLMC in 100 repetitions of the experiment. Data was again generated by the same model, with $n = 2048$. Each node represents one parameter, i.e., a relevant symbol (variable): if a node is included in the learned model, the corresponding symbol is included in the context. The nodes included in the true model are shaded. Note that the true model is not strictly hierarchical since all the individual effects on level four cancel, although there are nodes with effect on the fifth level. Overall, the transformed Lasso method has more false positives (it overfits), but fewer false negatives, than the VLMC method which only captures the most significant effects (it underfits; three out of six true effects are missed in all, or almost all, of the 100 repetitions). When looking at the negative log-likelihood of the learned models, Fig. 3, the VLMC method tends to give slightly better predictive probabilities than the transformed Lasso; for some other generating models the opposite holds. We will present more extensive experiments in future work.

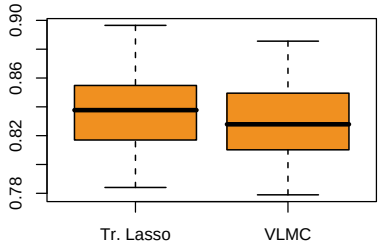

Fig. 3. Estimated per-symbol negative log-likelihood (log-loss) for the models learned by the transformed Lasso (*left*), and the VLMC method (*right*). The box shows the median (thick line) and quartiles; whiskers show min and max values; 100 repetitions. Smaller values are better.

VI. Future Work — Beyond Haar

We have used the Haar transformation as a prototypical example of a decomposition of Markov sources as a sum of components, each associated with a single parameter. The transformation idea is much more general: we can use other transformations, and the idea extends directly to other generalized linear models, not only logistic regression. Among alternative transformations, we mention the so called (Rademacher-)Walsh-Hadamard (WH) transformation, see [17]. It gives a decomposition of the parameter vector in terms of

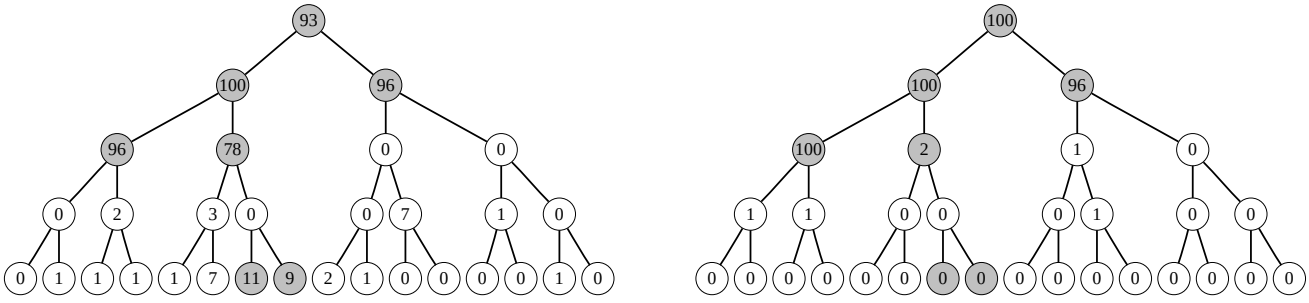

Fig. 4. Summary of learned models in Example 1, with $n = 2048$. Number of occurrences of (non-zero) parameters in 100 repetitions of the transformed Lasso (*left*) and VLMC (*right*). Each node in the tree corresponds to a parameter: e.g., the rightmost node on the second level (from top) corresponds to the effect of X_{i-2} given that $X_{i-1} = 1$. Only nodes up to level five are shown (maximum was $k = 7$). The shaded nodes are included in the true model.

XOR (parity) functions of increasing order on the indicator functions. Since ℓ_1 regularization can handle very high-dimensional problems, it is also possible to use overcomplete bases obtained by combining more than one complete bases.

The WH basis can be used to discover dependencies on possibly distant symbols in the past, cf. [18]. Both in the Haar and WH cases, the problem with allowing long distances is that the number of parameters grows exponentially in the distance. It is, however, possible to allow only some, say up to third order, cross-terms in the model. For instance, it is particularly simple (and common) to allow only *individual* effects of the symbols $X_{i-1}, X_{i-2}, \ldots, X_{i-k}$ by using k parameters/basis functions. Individual effects without any cross-terms are regularly used in (continuous-valued) autoregressive models, for which Lasso can be directly applied [19]. The transformed Lasso makes it simple to learn which effects, either individual or higher-order, are present in the data. This is useful from the point of view of understanding (modeling) the data, as well as prediction.

VII. CONCLUSIONS

The transformed Lasso approach provides efficient solutions (by convex optimization) for finding flexible context models. As an example, in this paper showed that it can be used for learning context trees (variable length Markov chains). The basic VLMC case is extensively studied in the information-theory literature, and thereby we believe that the transformed Lasso has more potential in more general contexts than those characterized by context trees. As noted above, this is achieved to some extent even by the Haar basis, which allows non-hierarchical structures where internal nodes in the context-tree are skipped. The generality of the approach will be more clear when using, for instance, the Walsh-Hadamard transformation.

In future work, we are planning to study algorithmic improvements in order to further reduce the computational complexity of the method. We will also consider the statistical properties, such as consistency and rate of convergence, related to various transformations. For instance, as explained above, it seems important to omit the normalizing coefficients in the Haar transformation to guarantee reliable model selection. Also, restrictions to hierarchical and bounded-order models are interesting from both theoretical and practical point of view.

ACKNOWLEDGMENTS

TR was supported in part by the Academy of Finland (project Modest). BY was supported in part by grants from MSRA and NSF DMS-0605165, NSFC (60628102), and an NSF CDI award.

REFERENCES

[1] J. Rissanen, "Universal coding, information, prediction, and estimation," *IEEE Trans. Inform. Theory*, vol. 30, pp. 629–636, 1983.
[2] M. Weinberger, J. Rissanen, and M. Feder, "A universal finite memory source," *IEEE Trans. Inform. Theory*, vol. 41, pp. 643–652, 1995.
[3] P. Bühlmann and A. Wyner, "Variable length Markov chains," *Ann. Statist.*, vol. 27, pp. 480–513, 1998.
[4] F. Willems, Y. Shtarkov, and T. Tjalkens, "The context-tree weighting method: Basic properties," *IEEE Trans. Inform. Theory*, vol. 41, pp. 653–664, 1995.
[5] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. on Scientific Computing*, vol. 20, pp. 33–61, 1998.
[6] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Royal Statist. Soc. B*, vol. 58, no. 1, pp. 267–288, 1996.
[7] M. Osborne, B. Presnell, and B. Turlach, "A new approach to variable selection in least squares problems," *J. Num. Analysis*, vol. 20, pp. 389–403, 2000.
[8] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Ann. Statist.*, vol. 32, no. 2, pp. 407–499, 2004.
[9] M. Y. Park and T. Hastie, "L1 regularization path algorithm for generalized linear models," *J. Royal Statist. Soc. B*, vol. 69, pp. 659–677, 2007.
[10] L. Meier, S. van de Geer, and P. Bühlmann, "The group lasso for logistic regression," *J. Royal Statist. Soc. B*, vol. 70, no. 1, pp. 53–71, 2008.
[11] G. McLachlan, *Discriminant Analysis and Statistical Pattern Recognition*, New York, 1992.
[12] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *J. Royal Statist. Soc. B*, vol. 68, no. 1, pp. 49–67, 2005.
[13] T. Hesterberg, N. H. Choi, L. Meier, and C. Fraley, "Least angle and ℓ_1 penalized regression: A review," *Statist. Surveys*, vol. 2, pp. 61–93, 2008.
[14] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight, "Sparsity and smoothness via the fused lasso," *J. Royal Statist. Soc. B*, vol. 67, no. 1, pp. 91–108, 2005.
[15] S. Mallat, *A Wavelet Tour of Signal Processing*. San Diego, CA: Academic Press, 1998.
[16] H. Zou, T. Hastie, and R. Tibshirani, "On the "degrees of freedom" of the Lasso," *Ann. Statist.*, vol. 35, no. 5, pp. 2173–2192, 2007.
[17] T. Beer, "Walsh transforms," *Am. J. Phys.*, vol. 49, no. 5, 1981.
[18] X. Zhao, H. Huang, and T. Speed, "Finding short DNA motifs using permuted Markov models," *J. Comp. Biol.*, vol. 12, no. 6, pp. 894–906, 2005.
[19] N.-J. Hsu, H.-L. Hung, and Y.-M. Chang, "Subset selection for vector autoregressive processes using Lasso," *Comp. Stat. & Data Analysis*, vol. 52, no. 7, pp. 3645–3657, 2008.