Structure Learning in Random Fields for Heart Motion Abnormality Detection

Mark Schmidt, Kevin Murphy
Computer Science Dept.
University of British Columbia
Vancouver, BC

Glenn Fung, Rómer Rosales
IKM CAD and Knowledge Solutions USA, Inc.
Siemens Medical Solutions
Malvern, PA 19355

Abstract

Coronary Heart Disease can be diagnosed by assessing the regional motion of the heart walls in ultrasound images of the left ventricle. Even for experts, ultrasound images are difficult to interpret leading to high intra-observer variability. Previous work indicates that in order to approach this problem, the interactions between the different heart regions and their overall influence on the clinical condition of the heart need to be considered. To do this, we propose a method for jointly learning the structure and parameters of conditional random fields, formulating these tasks as a convex optimization problem. We consider block-L1 regularization for each set of features associated with an edge, and formalize an efficient projection method to find the globally optimal penalized maximum likelihood solution. We perform extensive numerical experiments comparing the presented method with related methods that approach the structure learning problem differently. We verify the robustness of our method on echocardiograms collected in routine clinical practice at one hospital.

1. Introduction

We consider the task of detecting coronary heart disease (CHD) by measuring and scoring the regional and global motion of the left ventricle (LV) of the heart. CHD typically causes local segments of the LV wall to move abnormally. The LV can be imaged in a number of ways. The most common method is the echocardiogram – an ultrasound video of different 2-D cross-sections of the LV (see Figure 1 for an example). This paper focuses on the pattern recognition problem of classifying LV wall segments, and the heart as a whole, as normal or abnormal from an ultrasound sequence. The algorithms used for automatic detection, tracing and tracking of contours to extract features of the LV wall segments are described in [38].

Echocardiograms are notoriously difficult to interpret, and even the best of physicians can misdiagnose heart disease. Hence, there is a tremendous need for an automated *second-reader* system that can provide objective diagnostic assistance. Inter-observer studies have shown high intra-observer variability, evidencing how challenging the problem is in practice.

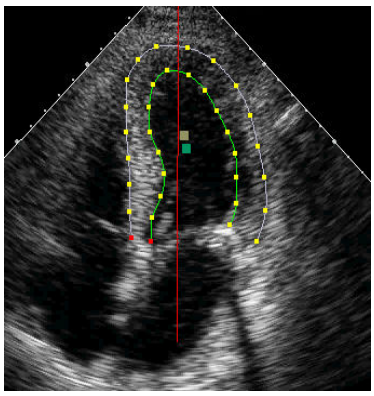

Figure 1. One frame/view from an LV ultrasound image clip. The contours delinate the walls of the left ventricular chamber in this particular view (one of three used). For each given view, these contours are used to track the movement of the LV wall segments and generate the features used to train our model.

From clinical knowledge it is known that the heart wall segments (specifically the myocardial LV segments) do not move independently, but they have an effect on each other. For example, an abnormal segment could be dragged in the right direction by its contiguous neighbors (e.g.; due to the muscle physiology), giving the false impression of being normal. The opposite can also occur, several segments may look abnormal but in reality there may be only one abnormal segment (potentially diseased). These effects may lead to correlations (both positive and negative) in the labels of the 16 LV segments. These would not be taken into account if the joint classification problem were split into 16 independent classification tasks.

We hypothesize that the interconected nature of the heart muscle can be more appropriately characterized by structured output models. We focus in particular on Conditional Random Fields (CRFs). CRFs are undirected graphical models that can be used to compactly represent con-

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

ditional probability distributions, $p(\mathbf{y}|\mathbf{x})$, where $\mathbf{y}$ are the labels (i.e., condition of the heart segments) and $\mathbf{x}$ are observed features (describing the motion of the segments). CRFs often outperform iid classifiers by taking into account dependencies between the labels. Another appealing aspect of CRFs in the context of classifying the segments of the LV, as opposed to many tasks where CRFs are applied, is the relatively small number of nodes in the graph. For 16 nodes, all computations in a joint CRF classification model are tractable (2^{16} combinations of labels can be enumerated for inference or calculation of the partition function in reasonable CPU time).

Usually the structure of CRFs is specified by hand. For example, it is often assumed to be a linear chain (for sequence labeling problems e.g., [20]) or a 2D lattice (for image processing problems e.g., [19]). However, in our heart wall motion analysis problem, it is not clear what graph structure to use. Recent work has examined learning tree-structure graphs and Directed Acyclic Graphs (DAGs) [30] trained on labels alone. These structures are acyclic and thus may not capture the complexity in the labels. Furthermore, it is not clear that generatively learning a graph structure on the labels alone will have good performance when used within a discriminative classifier. In this paper, we introduce a new approach for simultaneously learning both the structure and parameters of a CRF classifier based on block-L1 regularized optimization, and apply it to this challenging medical problem. Our efficient optimization algorithm for block-L1 regularized estimation may also be of use in other applications.

Ref	(G/D,D/U,Opt)	Method	Restrictions
[3]	(G ,U ,N)	Greedily add features	Thin
[8]	(G ,U/D,Y)	MinSpanTree	Tree struct
[11]	(G ,D ,Y)	Semi definite program	Fan-in
[34]	(G ,D ,N)	Greedy order search	Fan-in
[13]	(G ,D ,N)	Greedy DAG search	Fan-in
[7]	(G ,D ,Y)	Greedy Equiv Search	Fan-in
[17]	(G ,D ,Y)	Dynamic program	Exp time/space
[18]	(G ,U ,N)	Inductive logic program	Markov net
[25]	(G ,U ,Y)	L1MB	Gaussian
[36]	(G ,U ,Y)	L1MB	Binary
[22]	(G ,U ,Y)	L1RF + LBP	Binary
[15]	(G ,U ,Y)	MER + jtree	Bnry, Thin
[27]	(G ,U ,N)	Exhaustive search	-
[9]	(D ,D ,N)	Greedy DAG Search	-
[10]	(D ,D ,N)	Exhaustive search	-
[26]	(D ,U/D,N)	Submod-supermod opt.	TAN
[29]	(D ,U/D,N)	Greedily add best edge	TAN
[27]	(D ,U ,N)	Exhaustive search	-
This	(D ,U ,Y)	Block-L1 CRF	-

Table 1. Summary of some approaches for learning graphical model structure. First group: DAGs; second group: MRFs; third group: CRFs. We only consider structure learning, not parameter learning. Columns from left to right: G/D: G = generative, D = discriminative. D/U: U = undirected, D = directed. Opt: can the global optimum of the specified objective be obtained? Method: see text. LBP = loopy belief propagation; jtree= junction tree; MER = Maximum Entropy Relaxation Restrictions: fan-in = bound on possible number of parents (for DAG models); thin = low tree width; TAN = tree augmented network.

2. Structure Learning and L1-Regularization

We can categorize approaches for structure learning along several axes: (1) learning the topology based on the labels alone (a model of $p(\mathbf{y})$), or based on the features as well (a model of $p(\mathbf{y}|\mathbf{x})$); (2) learning directed graphs or undirected graphs[1]; (3) learning an arbitrary graph structure or restricting the graph in some way (e.g., to trees or thin junction trees). We summarize a variety of existing approaches along these dimensions in Table 1.

From Table 1, we see that there has been very little work on discriminative structure learning (learning the topology given $\mathbf{y}$ and $\mathbf{x}$). All previous work in this vein focuses on the special case of learning a structure that is useful for estimating a single variable, namely the class label [9, 10, 26, 29]. That is, these methods model dependencies between the observed inputs, but only have a single output. In contrast, in this paper, we consider classification with "structured output", i.e., with multiple dependent class labels by learning the structural dependencies between the outputs. We believe this is the first paper to address the issue of discriminative learning of CRF structure. We focus on undirected graphs of arbitrary topology with pairwise potentials and binary labels. (This assumption is for notational simplicity, and is not required by our methods.)

Recently, one of the most popular approaches to generative structure learning is to impose an L1 penalty on the parameters of the model, and to find the MAP parameter estimate. The L1 penalty forces many of the parameters, corresponding to edge features, to go to zero, resulting in a sparse graph. This was originally explored for modeling continuous data with Gaussian Markov Random Fields (MRFs) in two variants. In the *Markov Blanket* (MB) variant, the method learns a dependency network [12] $p(y_i|\mathbf{y}_{-i})$ by fitting d separate regression problems (independently regressing the label of each of the d nodes on all other nodes), and L1-regularization is used to select a sparse neighbor set [25]. Although one can show this is a consistent estimator of topology, the resulting model is not a joint density estimator $p(\mathbf{y})$ (or $p(\mathbf{y}|\mathbf{x})$ in the conditional variant we explore), and cannot be used for classification. In the *Random Field* (RF) variant, L1-regularization is applied to the elements of the precision matrix to yield sparsity. While the RF variant is more computationally expensive, it yields

[1] Generative models can be directed (Bayes nets) or undirected (MRFs), whereas discriminative models are usually undirected since discriminative directed models, such as [23], suffer from the "label bias" problem [20]. Trees and other chordal graphs can be directed or undirected without changing their expressive power.

both a structure and a parameterized model (while the MB variant yields only a structure). For modeling discrete data, analogous algorithms have been proposed for the specific case where the data is binary and the edges have Ising potentials ([36] present the discrete MB variant, while the discrete RF algorithm is presented in [22]). In this binary-Ising case, there is a 1:1 correspondence between parameters and edges, and this L1 approach is suitable. However, in more general scenarios (including any combination of multi-class MRFs, non-Ising edge potentials, or CRFs like in this paper), where many features are associated with each edge, *block-L1 methods* that jointly reduce *groups* of parameters to zero at the same time need to be developed in order to achieve sparsity.

Although such extensions were discussed in [22], there has been (as far we know) no attempt at formulating or implementing them. We believe that this is due to three (related) unsolved problems: (i) there are an enormous number of variables (and variable groups) to consider even for small graphs with a small number of features (ie. for n nodes with k states and p features per node, the number of groups is $O(n^2)$ and the number of variables is $O(kp^2)$), (ii) in the case of RF models the optimization objective function can be very expensive or intractable to evaluate (with a worst-case cost of $O(k^n)$), and (iii) existing block-L1 optimization strategies do not scale to this large number of variables (particulary when the objective function is expensive to evaluate). After reviewing CRFs and block-L1 formulations in Sect. 3 and 4, in Sect. 5 we will review existing block-L1 methods and then outline an algorithm takes advantage of recent advances in the optimization community and the structure of the problem in order to solve the problem efficiently.

3. Conditional Random fields

Definitions: In this paper we consider CRFs with pairwise potentials:

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{<ij>} \psi_{ij}(y_i, y_j, \mathbf{x}) \prod_i \psi_i(y_i, \mathbf{x}) \quad (1)$$

where $<ij>$ is a product over all edges in the graph, ψ_i is a node potential (local evidence term) and ψ_{ij} is an edge potential. For notational simplicity, we focus on binary states, $y_i \in \{1, 2\}$. We assume the node and edge potentials have the following form:

$$\psi_i(\cdot, \mathbf{x}) = \begin{pmatrix} e^{\mathbf{v}_{i,1}^T \mathbf{x}_i}, & e^{\mathbf{v}_{i,2}^T \mathbf{x}_i} \end{pmatrix} \quad (2)$$

$$\psi_{ij}(\cdot, \cdot, \mathbf{x}) = \begin{pmatrix} e^{\mathbf{w}_{ij,11}^T \mathbf{x}_{ij}} & e^{\mathbf{w}_{ij,12}^T \mathbf{x}_{ij}} \\ e^{\mathbf{w}_{ij,21}^T \mathbf{x}_{ij}} & e^{\mathbf{w}_{ij,22}^T \mathbf{x}_{ij}} \end{pmatrix} \quad (3)$$

where $\mathbf{x}_i = [1, \mathbf{g}, \mathbf{f}_i]$ are the node features, $\mathbf{x}_{ij} = [1, \mathbf{g}, \mathbf{f}_i, \mathbf{f}_j]$ are the edge features, with $\mathbf{g}$ being global features shared across nodes and $\mathbf{f}_i$ being the node's local features. We set $\mathbf{v}_{i,2} = 0$ and $\mathbf{w}_{ij,22} = 0$ to ensure identifiability, otherwise the model would be over-parameterized.[2]

Representation: For the optimization problems introduced here, it is more convenient to use an alternative representation. If we write $\boldsymbol{\theta} = [\mathbf{v}, \mathbf{w}]$ for all the parameters and $F(\mathbf{x}, \mathbf{y})$ for all the features (suitably replicated), we can write the model more succinctly as $p(\mathbf{y}|\mathbf{x}) = \frac{\exp(\boldsymbol{\theta}^T F(\mathbf{x}, \mathbf{y}))}{Z(\boldsymbol{\theta}, \mathbf{x})}$ where $Z(\boldsymbol{\theta}, \mathbf{x}) = \sum_{\mathbf{y}'} \exp(\boldsymbol{\theta}^T F(\mathbf{x}, \mathbf{y}'))$. The negative log-likelihood and gradient are now given by:

$$\mathrm{nll}(\boldsymbol{\theta}) = \sum_{n=1}^{N} -\boldsymbol{\theta}^T F(\mathbf{x}_n, \mathbf{y}_n) + \sum_{n=1}^{N} \log Z(\boldsymbol{\theta}, \mathbf{x}_n) \quad (4)$$

$$\nabla \mathrm{nll}(\boldsymbol{\theta}) = -\sum_n [F(\mathbf{x}_n, \mathbf{y}_n) - E_{\mathbf{y}'} F(\mathbf{x}_n, \mathbf{y}')], \quad (5)$$

where $E_{\mathbf{y}'} F(\mathbf{x}_n, \mathbf{y}') = \sum_{\mathbf{y}'} p(\mathbf{y}'|\mathbf{x}_n, \boldsymbol{\theta}) F(\mathbf{x}_n, \mathbf{y}')$ are the expectations for the features.

Tractability: One can show that this expectation factorizes according to the graph structure (see e.g., [32]). Nevertheless, computing the gradient is expensive, since it requires an inference (state estimation) algorithm. This takes $O(k^w)$ time, where w is the tree width of the graph and k is the number of labels for each y_i (we are assuming $k = 2$). For a chain $w = 2$. In practice we do not know the topology (we are learning it), and thus in general $w = d$, the number of nodes. There are three solutions to this: restrict the graph to have low tree width [3, 15]; use approximate inference, such as loopy belief propagation (used in [22]) or brief Gibbs sampling (used in [14]); or change the objective function to pseudo-likelihood [5]. The first alternative would restrict the type of graphs we can learn, making our approach rather limited. The other two alternatives do not limit the space of possible graphs, and will be compared in our experiments (along with the exact conditional). We will particularly focus on pseudo-likelihood (PL) as an alternative to the exact nll that greatly reduces the complexity of the optimization problem we propose, but maintains several appealing properties.

Pseudo-likelihood: PL is defined as $PL(\mathbf{y}^n|\mathbf{x}^n) = \prod_i p(y_i^n|\mathbf{y}_{n_i}^n, \mathbf{x}^n)$, where n_i are the neighbors of i in the graph, where $p(y_i^n|\mathbf{y}_{n_i}^n, \mathbf{x}^n) = \exp(\boldsymbol{\theta}_i^T \mathbf{F}_i(\mathbf{x}, \mathbf{y}))/Z_i$, where $\boldsymbol{\theta}_i = (\mathbf{v}_i, \{\mathbf{w}_{ij}\}_{j \in n_i})$ are the parameters for i's Markov blanket, Z_i is the local partition function, and $\mathbf{F}_i$ is the local feature vector. PL is known to be a consistent estimator of the parameters (as the sample size goes to infinity),

[2] Note that we can recover an MRF for representing the unconditional density $p(\mathbf{y}, \mathbf{x})$ by simply setting $\mathbf{x}_{ij} = 1$. In that case, the elements of $\mathbf{w}_{ij}$ will represent the unconditional potential for edge $i - j$. (In the MRF case, the ψ_i potentials are often locally normalized as well, but this is not required.) If in addition we require $\mathbf{w}_{ij,11} = \mathbf{w}_{ij,22} = w_{ij}$, and $\mathbf{w}_{ij,21} = \mathbf{w}_{ij,12} = -w_{ij}$, we recover an Ising model.

and is also convex (unlike the loopy belief propagation approximation to the likelihood used in the previous discrete L1-regularized RF model [22]). Furthermore, it only involves local partition functions, so it can be computed very efficiently ($O(d)$ in terms of d instead of $O(2^d)$ for the exact binary likelihood).

For our structure learning problem, inference is necessary at test time in order to compute marginals $p(y_i|\mathbf{x})$ or MAP estimates $\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} p(\mathbf{y}|\mathbf{x})$. Owing to the small number of nodes present, we will use exact inference in our experiments, but loopy belief propagation or other approximate inference procedures are again a possibility for larger graphs (as in [22]).

4. Block L1 regularizers

We formulate our regularized structure learning problem by placing an L2 regularizer on the local evidence parameters $\mathbf{v}$ (which do not affect the graph structure directly), and the critical regularizer $R(\mathbf{w})$ (affecting the learned structure) on the edge parameters $\mathbf{w}$:

$$J(\boldsymbol{\theta}) = \mathrm{nll}(\boldsymbol{\theta}) + \lambda_2||\mathbf{v}||_2^2 + \lambda_1 R(\mathbf{w}) \tag{6}$$

We consider nll to be either the exact negative log-likelihood or a suitable approximation like PL. We consider the following form for the edge (structural) regularizer $R(\mathbf{w})$:

$$R(\mathbf{w}) = \sum_{b=1}^{B} \left(\sum_{i \in b} |w_i|^{\alpha} \right)^{1/\alpha} = \sum_b ||\mathbf{w}_b||_\alpha \tag{7}$$

where $\mathbf{w}_b$ corresponds to parameter block b (we have one block per edge in the graph). If we use $\alpha = 1$, this degenerates into the standard L1/Lasso regularizer $R_1(\mathbf{w}) = ||\mathbf{w}||_1$ (we refer to this as L_1L_1). This non-differentiable problem can be solved efficiently using variants of the L-BFGS Quasi-Newton algorithm (see [1]), but does not yield sparsity at the block level. A common approach for imposing sparsity at the block level in order to force all the parameters in a block to go to zero is to use $\alpha = 2$, $R_2(\mathbf{w}) = \sum_b \sqrt{\sum_{i \in b} w_i^2}$. This is sometimes called the Group-Lasso[3], but we call it L_1L_2. This is also non-differentiable, and the equivalent constrained formulation results in a second order cone program (rather than linear constraints as in L_1L_1), which can be expensive to optimize. A more computationally appealing alternative is to use $\alpha = \infty$: $R_\infty(\mathbf{w}) = \sum_b \max_{i \in b} |w_i|$, which we will call L_1L_∞. This choice of α also yields sparsity at the block level [35], but as we will see in Sect. 5, this results in a linearly-constrained smooth objective that can be solved efficiently.

[3]In the Group-Lasso, the regularizer for each block is scaled propotional by its size. To simplify notation, we ignore this scale factor.

5. Optimization

We now consider how to minimize the two different regularized objectives defined in Eq. 6 for $\alpha = 2$ and $\alpha = \infty$ (ie. the choices that yield sparsity at the block-level). On its own the L_1L_2 regularizer is the objective (subject to linear constraints) in the famous sum-of-norms problem studied by Fermat (and subsequently many others) in the 1600s.Used as a regularizer for a twice-differentiable objective function, it can be optimized in a variety of ways. Block Coordinate Descent (BCD) methods have been proposed for this type objective function in the cases of linear [37] and logistic [24] regression. These strategies are very efficient when a relatively small of number of blocks are non-zero at the solution (and the blocks are reasonably independent), or in cases where the objective function (and optimization of a block of variables keeping the others fixed) can be done efficiently. These methods are not well suited for our objective, since we would like to explore models where hundreds of edges are active (or thousands for larger data sets), the objective function can be very expensive to evaluate (thus calculating nll a large number of times is not appealing), and by design there may exist strong correlations among the blocks. An alternative is the gradient projection method of [16] that is able to move a large number of variables simultaneously and thus can reduce the number of function evaluations required. However, it involves an expensive projection step that does not separate across blocks, and the use of the vanilla steepest descent direction results in slow convergence. Proposed primal-dual interior point methods (e.g., [33]) and path following [28] approaches require exact Hessians, which are intractable in our setting (not only to compute since they involve the joint distribution of all pairs of variables even if they are not adjacent, but to store given the large number of variables involved). In order to solve problems of this type, we approximated the L_1L_2 regularizer using the multi-quadric function $||\mathbf{w}||_2 \approx \sqrt{\mathbf{w}^T\mathbf{w} + \epsilon}$, (similar to [21]), and used a limited-memory BFGS algorithm to optimize this differentiable objective for a small positive ϵ. This is not especially efficient since the approximated curvature matrix is ill-conditioned numerically, but it did allow us to reach high accuracy solutions in our experiments (eventually). A previous algorithm has been proposed for optimizating a twice-differentiable function with L_1L_∞ regularization based on interior point methods [35]. However, this method require the Hessian (which is computationally intensive to both compute and store in our setting, as discussed above). We now propose a first-order method that does not need the Hessian, but still converges quickly to the optimal solution by moving all variables simultaneously along the projected gradient direction with a cleverly chosen step length. In contrast to the regular L1 objective function that is differentiable everywhere except at zero, the L_1L_∞ objective function is additionally non-differentiable when

there are ties between the maximum magnitude variables in a block. Since at the optimal solution we expect all variables in some blocks to be 0, this makes the application of many existing L1 optimization strategies (such as smoothing methods or the sub-gradient method of [22]) problematic. Rather than using a sub-gradient strategy (or an unconstrained smooth approximation), we convert the problem to a constrained optimization by reformulating in terms of auxiliary variables (one for each set) that are constrained to be the maximum value of a set. Since minimizing the block-L1 in the set $S = s_1,...,s_n$ is equivalent to minimizing the infinity norm $\|(s_1,...,s_n)\|_\infty = \max_i \{|s_i|\}$, we can add linear constraints and a linear term to yield the following equivalent mathematical program:

$$\begin{array}{c} \min_{\boldsymbol{\alpha},\mathbf{v},\mathbf{w}} \mathrm{nll}(\boldsymbol{\theta}) + \lambda_2 \|\mathbf{v}\|_2^2 + \lambda_1 \sum_s \alpha_s \\ st. \quad \forall_s \, \forall_k \in s \quad -\alpha_s \leq w_{sk} \leq \alpha_s \end{array} \tag{8}$$

where s indexes the blocks (edges). To describe our algorithm we will use $\mathbf{x}_k = \{\boldsymbol{\alpha}, \mathbf{w}, \mathbf{v}\}$ to denote the concatenation of all variables and $f(x_k)$ as the value of the objective function at iterate k. Our algorithm for solving this constrained optimization problem falls in the class of gradient-projection methods. A common variant of gradient-projection methods compute a direction of descent at iterate k by finding the Euclidean-norm projection of a scaled steepest descent direction onto the feasible set. Using Π to denote this projection, β as the scale factor for the steepest descent direction, and t as a step length chosen by a line search procedure, the iterates can be written as[4]: $\mathbf{x}_{k+1} = \mathbf{x}_k + t(\Pi(\mathbf{x}_k - \beta \nabla f(\mathbf{x}_k)) - \mathbf{x}_k)$, Unfortunately, there are 2 severe drawbacks of this type of approach: (i) in general the projection step involves solving a large Quadratic Program, and (ii) the use of the steepest descent direction results in slow convergence and an unnacceptably large number of function evaluations. We will address the latter problem first.

In [4], a variant of the steepest descent algorithm was proposed where the step length β along the steepest descent direction is chosen as in the inverse Raliegh quotient $\beta = \frac{\mathbf{s}^T\mathbf{s}}{\mathbf{s}^T\mathbf{y}}$ in order to satisfy the secant equation (where $\mathbf{s} = \mathbf{x}_k - \mathbf{x}_{k-1}$, $\mathbf{y} = \nabla f(\mathbf{x}_k) - \nabla f(\mathbf{x}_{k-1})$). Referred to as the 'Barzilai and Borwein' (BB) algorithm after its authors, this method has received increased attention in the optimization community since global convergence under a non-monotone line search was proved in [31], which also showed that this simple and memory-efficient method is competitive computationally with more complex approaches. Due to its use of the steepest descent direction, the non-monotone BB step can also be used to significantly speed up the convergence of Gradient Projection algorithms, without an increase in the cost of the projection (since the projection can be still be done under the Euclidean norm). This is often referred to as the 'Spectral Projected Gradient' (SPG) algorithm [6]. In this vein, we use a non-monotone Armijo line search [2] to find a t that satisfies the following condition (using sufficient decrease parameter $c = 10^{-4}$ over the the last $p = 10$ steps, and $\mathbf{d} \triangleq \Pi(\mathbf{x}_k - \beta \nabla f(\mathbf{x}_k)) - \mathbf{x}_k$)

$$f(\mathbf{x}_k + t\mathbf{d}) \leq \max_{i=k-p:k} f(\mathbf{x}_i) + ct \nabla f(\mathbf{x}_k)^T \mathbf{d} \tag{9}$$

Using an SPG strategy yields an algorithm that converges to the optimal solution after a relatively small number of function evaluations. In our case the projector operator $\Pi(\mathbf{x})$ onto the convex feasible set $\mathcal{F} = \{\{\boldsymbol{\alpha}, \mathbf{w}, \mathbf{v}\} \; | \forall_k \in s \quad -\alpha_s \leq w_{sk} \leq \alpha_s\}$ is defined as $\mathbf{x}^* = \min_{\mathbf{x} \in \mathcal{F}} \|\mathbf{x} - \mathbf{u}\|_2^2$ which may be expensive to solve at each iteration for large-scale problems. However, the projection is separable across groups, which means we just have to solve the following for each $(\mathbf{w}_s, \alpha_s)$ independently, rather than jointly (the projection does not change v):

$$\min_{\mathbf{w}', \alpha'} \|(\mathbf{w}', \alpha') - (\mathbf{w}_s, \alpha_s)\|_2^2 \; st. \; \forall_i \; -\alpha' \leq w_i' \leq \alpha' \tag{10}$$

Thus, we can efficiently compute the optimal projection by a solving a small linearly constrained problem for each group (an interior point method was used for this purpose). We summarize the overall algorithm in Algorithm 1.

6. Experimental Results

We have experimentally compared an extensive variety of approaches to learning the CRF graph structure and the associated parameters. Below we divide up the approaches into several groups:

Fixed Structures: We learn the parameters of a CRF with a fixed structure (using L-BFGS). We considered an *Empty* structure (corresponding to iid Logistic Regression), a *Chain* structure (as in most CRF work), a *Full* structure (assuming everything is dependent), and the *True* structure. For the synthetic experiments, the *True* structure was set to the actual generating structure, while for the Heart experiments we generated a *True* structure by adding edges between all nodes sharing a face in the heart diagram, constructed by expert cardiologists, from [30].

Generative Structures: We learn a model structure based on the labels alone, and then learn the parameters of a CRF with this fixed structure. We considered block-L1 methods for $\alpha = \{1, 2, \infty\}$ for both the *MB* and *RF* variants. We also considered the two non-L1 generative models from [30], finding the optimal *Tree* (using the Chow-Liu algorithm) and *DAG-Search* with greedy hill-climbing.

[4] It is possible to fix t at 1 and perform the line search along the projection arc by varying β. This results in quicker identification of the active set but is innefficient for our purposes since it involves multiple projections in the line search.

Algorithm 1 pseudo-code for SPG to solve optimization problem (8)

Given an initial point $\mathbf{x}_0$
while 1 **do**
  Compute $f(\mathbf{x}_k)$ and $\nabla f(\mathbf{x}_k)$
  if $k = 0$ **then**
    $\beta = 1$
  else {Compute the BB quotient}
    $\mathbf{s} = \mathbf{x}_k - \mathbf{x}_{k-1}$
    $\mathbf{y} = \nabla f(\mathbf{x}_k) - \nabla f(\mathbf{x}_{k-1})$
    $\beta = \mathbf{s}^T\mathbf{s}/\mathbf{s}^T\mathbf{y}$
  end if
  $\bar{\mathbf{x}} = \mathbf{x}_k - \beta\nabla f(\mathbf{x}_k)$
  for each group s **do**
    Solve problem (10) to calculate the projection $\Pi(\bar{\mathbf{w}}_s, \bar{\alpha}_s)$
  end for
  Compute the descent direction: $\mathbf{d} = \Pi(\bar{\mathbf{x}}) - \mathbf{x}_k$
  if $\nabla f(\mathbf{x}_k)^T\mathbf{d} \leq \epsilon$ **then**
    break
  end if
  Compute the step length t to satisfy (9)
  Compute the new iterate $\mathbf{x}_{k+1} = \mathbf{x}_k + t\mathbf{d}$
  $k = k + 1$
end while

Discriminative Structures: Finally we explored the main contribution of this paper, conditional L1-based structure learning for $\alpha = \{1, 2, \infty\}$. In the MB variant, the structure is conditionally learned first, then the CRF is trained with the fixed structure. In the RF variant, the structure and the parameters are learned simultaneously.

6.1. Synthetic Data

To compare methods and test the effects of both discriminative structure learning and approximate inference for training, we created a synthetic dataset from a small (10-node) CRF (we discuss larger models below). We used 10 local features for each node (sampled from a standard Normal) plus a bias term. We chose the graph structure randomly, including each edge with probability $p_e = 0.5$. Similarly, we sampled random node weights $\mathbf{v}_i \sim \mathcal{N}(0, \sqrt{2})$, and edge weights $\mathbf{w}_{ij} \sim U(-b, b)$, where $b \sim \mathcal{N}(0, \sqrt{2})$ for each edge. We drew 500 training samples and 1000 test samples from the exact distribution $p(\mathbf{y}|\mathbf{x})$.

In all models, we impose an L2 penalty on the node weights, and we also impose an L2 penalty on the edge weights for all models that do not use L1 regularization of the edge weights. For each of the resulting 24 types of models compared, the scale of these two regularization parameters is selected by cross-validation on the training set. In our experiments, we explored 10 different permutations of training and testing instances in order to quantify variation in the performance of the methods. For testing the quality of the models, we computed the classification error associated with the exact marginals $p(y_i|\mathbf{x})$. We compared learning with Pseudolikelihood (PL), Loopy Belief Propagation (LBP), and Exact inference.

Type		Random Field		
		PL	LBP	Exact
Fixed	Empty	1.00-1.00	1.00-1.00	1.00-1.00
	Chain	0.84-0.89	0.84-0.88	0.84-0.88
	Full	0.34-0.39	0.29-0.32	0.29-0.31
	True	0.09-0.13	0.00-0.05	0.00-0.05
Generative Non-L1	Tree	0.68-0.72	0.67-0.69	0.67-0.69
	DAG	0.81-0.85	0.78-0.83	0.78-0.83
Generative-L1	L1-L1	0.56-0.69	0.59-0.68	0.56-0.68
	L1-L2	0.58-0.70	0.60-0.70	0.60-0.69
	L1-Linf	0.57-0.69	0.58-0.70	0.51-0.67
Discriminative-L1	L1-L1	0.34-0.37	0.22-0.27	0.21-0.26
	L1-L2	0.04-0.08	0.00-0.02	0.00-0.01
	L1-Linf	0.12-0.15	0.06-0.09	0.05-0.09

Table 2. 25-75% Relative classification error rates (lower is better) on a synthetic 10-node CRF.

In Table 2, we show the relative classification error rate of different methods on the test set. More precisely, we show the distribution of $(E(m) - \min(E(:,t)))/(\max(E(:,t)) - \min(E(:,t)))$, where $E(m)$ is the number of classification errors made by method m on trial t. Although not necessary for the synthetic data, we use this measure since the Heart data examined next is a relatively small data set with class imbalance, and even though the ranking of the methods is consistent across trials, the particular data split on a given trial represents a confounding factor that obscures the relative performance of the methods. We summarize this distribution in terms of its interquartile range (a measure of the width of the central 50% interval of the distribution); this is a more robust summary than the standard mean and standard deviation. Thus the best possible score is 0.00–0.00, and the worst is 1.00–1.00.

The results show several broad trends: (a) PL and LBP are almost as good as exact likelihood, (b) discriminatively learned structures outperform generatively learned structures, (c) any kind of structure is better than no structure at all, (d) both block L1 methods outperform plain L1 in the discriminative case and (e) in the generative case, block L1 and plain L1 are very similar (since there are only three features per edge). We have also found that the MB and RF techniques are similar in performance, although we omit these results due to lack of space. Results on other synthetic data sets yield qualitatively similar conclusions, with one exception: on some data sets LBP produced results that were much worse than PL or Exact training (we suspect this may be due to non-convexity or non-convergence of the approximate inference on non-tree structures).

6.2. Heart Motion Abnormality Detection

The data consists of 345 cases for which we have associated images as well as ground truth; all of which were generated using pharmacological stress, which allows the physician to control the amount of stress a patient experiences. All the cases have been labeled at the heart wall segment level by a group of trained cardiologists. According to standard protocol, there are 16 LV heart wall segments. Each of the segments were ranked from 1 to 5 according to its movement. For simplicity, we coverted the labels to a binary (1 = normal, 2 - 5 = abnormal) for all of the tests we will describe (classes 3 to 5 are severely under-represented in the data).

For all our models, we used 19 local image features for each node calculated from the tracked contours (shown in Fig. 1). Among these features we have: local ejection fraction ratio, radial displacement, circumferential strain, velocity, thickness, thickening, timing, eigenmotion, curvature, and bending energy. We also used 15 global image features, and one bias term. Thus, the full heart wall motion model had 120 groups, and more than 20,000 features to choose from. We used $2/3$ of the data for training and hyper-parameter tuning, and $1/3$ of the data for testing (across 10 different splits). We trained various models using PL and tested them using exact inference. In Table 3, we show results for relative classification accuracy on the test set at the segment level and the heart level (the heart level decision is made by cardiologists by testing whether two or more segments are abnormal). Like in the previous table, these results show relative accuracy; thus best and worst possible scores are 0.00–0.00 and 1.00–1.00 respectively.

We see that the discriminative L_1L_∞ method performs among the best at the segment level (achieving a median *absolute* classification accuracy of 0.92), and is typically the best method at the important heart-level prediction task (achieving a median *absolute* accuracy of 0.86 and the lowest error rate at this task in 9 out of the 10 trials). It outperforms Chow-Liu and DAG-search, the best techniques previously used in [30]. We also tested using LBP for learning, but learning with LBP typically lead to parameters where the algorithm would not converge and lead to poor results.

6.3. Scaling up to Larger Problems

While our target application had a large number of features, it only had 16 nodes. However, our algorithm allows scaling to much larger graphs. To illustrate this, we compared the runtimes for training CRFs with L2-regularization (using L-BFGS), L1-regularization (using bound-constrained L-BFGS), and the L_1L_∞ -regularization (using our proposed algorithm) with pseudo-likelihood on larger graphs in order to reach an optimality tolerance of 10^{-7} (an accuracy much lower than typically needed in practice). For a fully connected 100-node CRF with 10 features per node (resulting in 4950 groups and a total of $169,000$ variables), the L2-regularized optimizer required about 6.5 min., the L1-regularized optimizer took about 4 min., while our L_1L_∞ -regularized optimizer took approximately 25 min. While this indicates very good scaling given the problem size, the difference can be attributed to 2 factors: (i) the Barzilai-Borwein steps require a larger number of iterations to converge than (bound-constrained) L-BFGS (which cannot be applied to block-L1 problems), and (ii) the expense of solving the thousands of projection problems at each iteration. The main factor to be considered for scaling to even larger problems is in fact not the number of nodes, but the number of edges that must be considered (since there are $O(d^2)$ possible edges for d nodes). The method can be further sped up by two natural approaches: parallelization (of function evaluations/projections), and restriction of the edge set considered (eg. by running an MB algorithm to prune edges before running the RF algorithm).

Type		Segment	Heart
Fixed	Empty	0.71-1.00	0.50-1.00
	Chain	0.36-0.75	0.50-1.00
	Full	0.29-0.55	0.33-0.50
	True	0.42-0.67	0.50-0.75
Generative Non-L1	Tree	0.33-0.89	0.50-1.00
	DAG	0.50-0.89	0.50-1.00
Generative-L1	L1-L1	0.27-0.50	0.50-0.67
	L1-L2	0.25-0.56	0.33-0.67
	L1-Linf	0.18-0.42	0.50-0.67
Discriminative-L1	L1-L1	0.50-0.88	0.83-1.00
	L1-L2	0.18-0.56	0.33-0.50
	L1-Linf	0.00-0.25	0.00-0.00

Table 3. 25-75% Relative classification error rates (lower is better) for AWMA at both the segment level and the heart level. The model was trained using PL, and tested using exact inference.

7. Conclusions and future work

We have developed a general method for learning (sparse) graph structures of general discriminative models via block-L1 regularization. The formulation involves casting the task as a convex optimization problem. In order to make it possible to use the proposed L_1L_∞ regularization, we introduced a new efficient approach to finding the global minimum of the resulting objective function, in particular for cases in which the Hessian is intractable to compute/store using standard methods.

Through experimental comparisons, we have demonstrated that this is an effective method for approaching our problem of segment/heart level classification from ultrasound video. We have shown that methods that model dependencies between labels outperform iid classifiers, and methods that learn the graph structure discriminatively outperform those that learn it in a non-discriminative manner.

We also provided an improved probablisitic model that

addresses the task of building a real-time application for heart wall motion analysis with the potential to make a significant impact in clinical practice. These encouraging results can also help less-experienced cardiologists improve their diagnostic accuracy; the agreement between less-experienced cardiologists and experts is often below 50%.

References

[1] G. Andrew and J. Gao. Scalable training of L1-regularized log-linear models. In *ICML*, 2007.

[2] L. Armijo. Minimization of functions having Lipschitz-continuous first partial derivatives. *Pacific J. of Mathematics*, 16:1–3, 1966.

[3] F. Bach and M. Jordan. Thin junction trees. In *NIPS*, 2001.

[4] J. Barzilai and J. Borwein. Two point step size gradient methods. *IMA J. of Numerical Analysis*, 8:141–148, 1988.

[5] J. Besag. Efficiency of pseudo-likelihood estimation for simple Gaussian fields. *Biometrika*, 64:616–618, 1977.

[6] E. G. Birgin, J. M. Martinez, and M. Raydan. Nonmonotone spectral projected gradient methods on convex sets. *SIAM J. on Optimization*, 10(4):1196–1211, 2000.

[7] D. M. Chickering. Optimal structure identification with greedy search. *JMLR*, 3:507–554, 2002.

[8] C. K. Chow and C. N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Trans. on Info. Theory*, 14:462–67, 1968.

[9] D. Grossman and P. Domingos. Learning Bayesian Network Classifiers by Maximizing Conditional Likelihood. In *ICML*, 2004.

[10] Y. Guo and R. Greiner. Discriminative Model Selection for Belief Net Structures. In *AAAI*, 2005.

[11] Y. Guo and D. Schuurmans. Convex Structure Learning for Bayesian Networks: Polynomial Feature Selection and Approximate Ordering. In *UAI*, 2006.

[12] D. Heckerman, D. Chickering, C. Meek, R. Rounthwaite, and C. Kadie. Dependency networks for density estimation, collaborative filtering, and data visualization. *JMLR*, 1:49–75, 2000.

[13] D. Heckerman, D. Geiger, and M. Chickering. Learning Bayesian networks: the combination of knowledge and statistical data. *Machine Learning*, 20(3):197–243, 1995.

[14] G. Hinton. Training products of experts by minimizing contrastive divergence. *N. Comput.*, 14:1771–1800, 2002.

[15] J. Johnson, V. Chandrasekaran, and A. Willsky. Learning Markov Structure by Maximum Entropy Relaxation. In *AI/Statistics*, 2007.

[16] Y. Kim, J. Kim, and Y. Kim. Blockwise sparse regression. *Statistica Sinica*, 16(2):375–390, 2006.

[17] M. Koivisto and K. Sood. Exact Bayesian structure discovery in Bayesian networks. *JMLR*, 5:549–573, 2004.

[18] S. Kok and P. Domingos. Learning the Structure of Markov Logic Networks. In *ICML*, 2005.

[19] S. Kumar and M. Hebert. Discriminative random fields: A discriminative framework for contextual interaction in classification. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2003.

[20] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, 2001.

[21] S. Lee, H. Lee, P. Abbeel, and A. Ng. Efficient L1 Regularized Logistic Regression. In *AAAI*, 2006.

[22] S.-I. Lee, V. Ganapathi, and D. Koller. Efficient structure learning of Markov networks using L1-regularization. In *NIPS*, 2006.

[23] A. McCallum, D. Freitag, and F. Pereira. Maximum Entropy Markov Models for Information Extraction and Segmentation. In *ICML*, 2000.

[24] L. Meier, S. van de Geer, and P. Buhlmann. The group lasso for logistic regression. Technical Report 131, ETH Seminar fur Statistik, 2006.

[25] N. Meinshausen and P. Buhlmann. High dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, 34:1436–1462, 2006.

[26] M. Narasimhan and J. Bilmes. A Supermodular-Submodular Procedure with Applications to Discriminative Structure Learning. In *UAI*, 2005.

[27] S. Parise and M. Welling. Structure learning in Markov Random Fields. In *NIPS*, 2006.

[28] M. Y. Park and T. Hastie. Regularization path algorithms for detecting gene interactions. Technical report, Stanford, 2006.

[29] F. Pernkopf and J. Bilmes. Discriminative versus Generative Parameter and Structure Learning of Bayesian Network Classifiers. In *ICML*, 2005.

[30] M. Qazi, G. Fung, S. Krishnan, R. Rosales, H. Steck, B. Rao, and D. Poldermans. Automated Heart Wall Motion Abnormality Detection from Ultrasound Images Using Bayesian Networks. In *Intl. Joint Conf. on AI*, 2007.

[31] M. Raydan. The barzilai and borwein gradient method for the large scale unconstrained minimization problem. *SIAM J. on Optimization*, 7(1):26–33, 1997.

[32] F. Sha and F. Pereira. Shallow parsing with conditional random fields. In *Proc. HLT-NAACL*, 2003.

[33] T. Simila and J. Tikka. Input selection and shrinkage in multiresponse linear regression. *Computational Statistics and Data Analysis*, 2007. To appear.

[34] M. Teyssier and D. Koller. Ordering-based search: A simple and effective algorithm for learning bayesian networks. In *UAI*, pages 584–590, 2005.

[35] B. Turlach, W. Venables, and S. Wright. Simultaneous variable selection. *Technometrics*, 47(3):349–363, 2005.

[36] M. Wainwright, P. Ravikumar, and J. Lafferty. Inferring graphical model structure using ℓ_1-regularized pseudo-likelihood. In *NIPS*, 2006.

[37] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *J. Royal Statistical Society, Series B*, 68(1):49–67, 2006.

[38] X. S. Zhou, D. Comaniciu, and A. Gupta. An information fusion framework for robust shape tracking. *TPAMI*, 27, NO. 1:115 – 129, January 2005.