Sparse Gaussian Conditional Random Fields: Algorithms, Theory, and Application to Energy Forecasting

Matt Wytock MWYTOCK@CS.CMU.EDU
J. Zico Kolter ZKOLTER@CS.CMU.EDU
Carnegie Mellon University, Pittsburgh PA

Abstract

This paper considers the sparse Gaussian conditional random field, a discriminative extension of sparse inverse covariance estimation, where we use convex methods to learn a high-dimensional conditional distribution of outputs given inputs. The model has been proposed by multiple researchers within the past year, yet previous papers have been substantially limited in their analysis of the method and in the ability to solve large-scale problems. In this paper, we make three contributions: 1) we develop a second-order active-set method which is several orders of magnitude faster than previously proposed optimization approaches for this problem, 2) we analyze the model from a theoretical standpoint, improving upon past bounds with convergence rates that depend logarithmically on the data dimension, and 3) we apply the method to large-scale energy forecasting problems, demonstrating state-of-the-art performance on two real-world tasks.

1. Introduction

Sparse inverse covariance estimation using ℓ_1 methods (Banerjee et al., 2008), also known as the graphical lasso (Friedman et al., 2008), enables convex learning of high-dimensional undirected graphical models. These methods estimate the inverse covariance of a zero-mean Gaussian distribution while penalizing the ℓ_1 norm of the off-diagonal entries; since the entries in the inverse covariance correspond to edges in a Gaussian Markov random field, this method learns a sparsely connected graphical model. In recent years, many algorithms have been proposed for this problem, including projected gradient methods (Duchi et al., 2008), smoothed optimization (Lu, 2009), alternating linearization methods (Scheinberg et al., 2010), and quadratic approximation (Hsieh et al., 2011).

However, in many prediction tasks we may not want to model correlations between input variables. This is the familiar generative/discriminative contrast in machine learning (Ng & Jordan, 2002), where it has been repeatedly observed that in terms of predictive accuracy, discriminative approaches can be superior (Sutton & McCallum, 2012). This has lead several researchers within the past year to (independently) propose a generalization of the Gaussian MRF, which we refer to as the sparse Gaussian conditional random field (CRF), that allows for discriminative modeling between input and output variables (Sohn & Kim, 2012), (Yuan & Zhang, 2012), and our own work in (Wytock & Kolter, 2012).[1] Although previous papers all showed significant promise to the model, they employed off-the-shelf optimization methods (significantly limiting the size of potential applications) and/or had theoretical results that did not fully highlight the advantages of sparsity.

In this paper, we make three contributions. First, we develop a specialized second-order active set method for estimating sparse Gaussian CRF parameters, which we show to be several orders of magnitude faster than previously proposed algorithms. Second, we develop convergence bounds for the algorithm that establish conditions for exact recovery of underlying models, with rates that specifically highlight the graph degree, improving upon the results in (Yuan & Zhang, 2012) in many settings. Third, we present extensive experimental results on large-scale synthetic data and

Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28. Copyright 2013 by the author(s).

[1] While these formulations were developed independently, they are mathematically identical, and so the model should be credited to (Sohn & Kim, 2012) as the first source, which termed the model "Sparse Conditional Gaussian Graphical Model". However, in this paper we refer to the model as the sparse Gaussian CRF, as the discriminative setting coincides with the standard notion of a conditional random field.

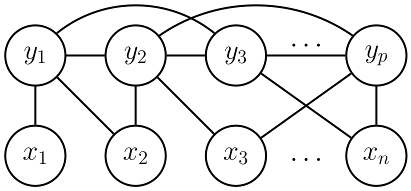

Figure 1. Illustration of sparse Gaussian CRF model.

two real-world energy forecasting tasks. Here we show improvement over state-of-the-art methods for wind power and electrical demand forecasting; these problems are of substantial practical interest, as even small advances in forecasting accuracy can have notable impact on the efficiency and costs of large power systems.

2. The sparse Gaussian CRF model

Let $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^p$ denote input and output variables for a prediction task. A Gaussian CRF is a log-linear model with

$$p(y|x; \Lambda, \Theta) = \frac{1}{Z(x)} \exp\left\{-y^T \Lambda y - 2x^T \Theta y\right\} \qquad (1)$$

where the quadratic term models the conditional dependencies of y and the linear term models the dependence of y on x. The model is parameterized by $\Lambda \in \mathbb{R}^{p \times p}$, which corresponds to the inverse covariance matrix, and $\Theta \in \mathbb{R}^{n \times p}$, which maps the inputs to the outputs; an illustration of the model is shown in Figure 1. Since the CRF is a Gaussian distribution with mean $-\Lambda^{-1}\Theta^T x$, the partition function is given by

$$\frac{1}{Z(x)} = c|\Lambda| \exp\left\{-x^T \Theta \Lambda^{-1} \Theta^T x\right\}. \qquad (2)$$

For m data samples, arranged as the rows of $X \in \mathbb{R}^{m \times n}$ and $Y \in \mathbb{R}^{m \times p}$, the negative log-likelihood $f(\Lambda, \Theta) = -\log p(Y|X; \Lambda; \Theta)$ is given by

$$f(\Lambda, \Theta) = -\log|\Lambda| + \operatorname{tr}\left(S_{yy}\Lambda + 2S_{yx}\Theta + \Lambda^{-1}\Theta^T S_{xx}\Theta\right) \qquad (3)$$

(omitting the constant term c term), where the S terms are empirical covariances

$$S_{yy} = \frac{1}{m} Y^T Y, \;\; S_{yx} = \frac{1}{m} Y^T X, \;\; S_{xx} = \frac{1}{m} X^T X. \quad (4)$$

Without regularization, it is straightforward to verify that this optimization problem is simply a reparameterization of the least squares problem. We can additionally add ℓ_2 regularization by adding λ_2 to the diagonal elements of S (formally, this corresponds to a Normal-Wishart prior on Λ and the columns of Θ), but again this just corresponds to the regularized least-squares solution. However, the total number of parameters in this problem (for estimating both Θ and Λ) is $np + p(p+1)/2$, and thus model can overfit when the number of examples m is relatively small.

To address this concern, we regularize the maximum likelihood estimate by adding ℓ_1 regularization to Θ and the off-diagonal elements of Λ; since the ℓ_1 norm encourages sparsity of the parameters, this directly corresponds to learning a sparse set of edges in our graphical model. Our final optimization problem is then given by minimizing the composite objective

$$\underset{\Lambda, \Theta}{\text{minimize}}\, f(\Lambda, \Theta) + \lambda(\|\Lambda\|_{1,\star} + \|\Theta\|_1) \qquad (5)$$

where $\|\cdot\|_1$ denotes the elementwise ℓ_1 norm, $\|\cdot\|_{1,\star}$ denotes the elementwise ℓ_1 norm on off-diagonal entries, and $\lambda \in \mathbb{R}_+$ is the regularization parameter.[2] This is a convex objective, following from the convexity of the ℓ_1 norm and the fact that the log-partition function of an exponential family graphical model is concave. Furthermore, the gradients of f are given by

$$\begin{aligned} \nabla_\Lambda f(\Lambda, \Theta) &= S_{yy} - \Lambda^{-1} - \Lambda^{-1}\Theta^T S_{xx}\Theta\Lambda^{-1} \\ \nabla_\Theta f(\Lambda, \Theta) &= 2S_{xy} + 2S_{xx}\Theta\Lambda^{-1}, \end{aligned} \qquad (6)$$

which in previous work has motivated the use of first-order non-smooth optimization methods.

3. Optimization

Previous work on the sparse Gaussian CRF (SGCRF) model has proposed using off-the-shelf algorithms to solve the above optimization problem, including orthantwise quasi-Newton methods (Sohn & Kim, 2012) (specifically, the OWL-QN method of (Andrew & Gao, 2007)), and accelerated proximal gradient methods (Yuan & Zhang, 2012) (specifically, the FISTA algorithm of (Beck & Teboulle, 2009)). These methods are attractive due to their simplicity, and since the gradients can be efficiently computed using (6). Unfortunately, the algorithms still suffer from relatively slow convergence (even though they are faster than many alternative non-smooth first-order methods), and thus quickly become computationally impractical for large output and input dimensions.

In this section, we propose a new second-order active set method for solving the sparse Gaussian CRF. Such algorithms have previously been applied to the Gaussian MRF (Hsieh et al., 2011; Olsen et al., 2012), and a general analysis of such methods (showing quadratic convergence) is presented in (Tseng & Yun, 2009). The

[2] It is also possible to introduce different regularization parameters for Λ and Θ, though we have found through our experiments that the optimal settings for these regularization parameters are typically quite similar, so we use only one for simplicity.

method here largely mirrors in the approach in (Hsieh et al., 2011) for the Gaussian MRF, but the precise formulation is significantly more involved, owing to the complexity of gradient term of the $\Lambda^{-1}\Theta^T S_{xx}\Theta$ term in the likelihood. Despite being a second-order method, we show that the resulting algorithm is faster (to reach any accuracy) than previously proposed approaches, and several orders of magnitude faster at achieving solutions to high accuracy.

3.1. A second-order active set approach

The basic idea of our method is to iteratively form a second-order approximation to the objective function (without the ℓ_1 regularization term), and then solve an ℓ_1 regularized quadratic program to find a regularized analog of the Newton step. In general notation, to minimize some objective $f(x) + \lambda\|x\|_1$, we form the Taylor expansion

$$f(x+\Delta) \approx g(\Delta) \equiv f(x) + \nabla_x f(x)^T \Delta + \frac{1}{2}\Delta^T \nabla_x^2 f(x)\Delta \tag{7}$$

where $\nabla_x f(x)$ and $\nabla_x^2 f(x)$ denote the gradient and Hessian respectively. To compute the regularized Newton step d, we solve

$$d = \arg\min_{\Delta} g(\Delta) + \lambda\|x + \Delta\|_1 \tag{8}$$

and update the parameters $x \leftarrow x + \alpha d$ for some stepsize α, determined by backtracking line search.

In our setting, precise formulations of the gradient and Hessian terms are cumbersome, due to the fact that all parameters involved are matrices, but we can concisely express this second order Taylor expansion using differentials. In particular, (see Appendix A for a full derivation) the second order Taylor expansion is

$$\begin{aligned} f(\Lambda + \Delta_\Lambda, \Theta + \Delta_\Theta) &\approx g(\Delta_\Lambda, \Delta_\Theta) \equiv f(\Lambda, \Theta) + \\ &\mathrm{tr} S_{yy}\Delta_\Lambda + 2\mathrm{tr} S_{yx}\Delta_\Theta - \mathrm{tr}\Lambda^{-1}\Delta_\Lambda + \\ &2\mathrm{tr}\Lambda^{-1}\Theta^T S_{xx}\Delta_\Theta - \mathrm{tr}\Lambda^{-1}\Theta^T S_{xx}\Theta\Lambda^{-1}\Delta_\Lambda + \\ &\mathrm{tr}\Lambda^{-1}\Delta_\Lambda\Lambda^{-1}\Theta^T S_{xx}\Theta\Lambda^{-1}\Delta_\Lambda + \frac{1}{2}\mathrm{tr}\Lambda^{-1}\Delta_\Lambda\Lambda^{-1}\Delta_\Lambda + \\ &\mathrm{tr}\Lambda^{-1}\Delta_\Theta^T S_{xx}\Delta_\Theta - 2\mathrm{tr}\Lambda^{-1}\Delta_\Lambda\Lambda^{-1}\Theta^T S_{xx}\Delta_\Theta. \end{aligned} \tag{9}$$

As above, we compute the Newton steps D_Λ, D_Θ by

$$\begin{aligned} D_\Lambda, D_\Theta = \arg\min_{\Delta_\Lambda, \Delta_\Theta} & \, g(\Delta_\Lambda, \Delta_\Theta) + \\ & \lambda\left(\|\Lambda + \Delta_\Lambda\|_{1,\star} + \|\Theta + \Delta_\Theta\|_1\right) \end{aligned} \tag{10}$$

where we use a coordinate descent algorithm to optimize this ℓ_1 regularized QP. Since ℓ_1 regularization on the Newton direction tends to push the Newton updates in a direction that increases sparsity and since the line search provably converges to step sizes with $\alpha = 1$ (Tseng & Yun, 2009), the number of nonzero elements tends to increase as the optimization progresses. For the line search, we additionally need to ensure that Λ is positive definite, which we ensure by a common technique of simply defining $-\log|X|$ to be infinite if $X \not\succ 0$. A generic pseudo-code description of the algorithm is shown in Algorithm 3.1, and a more detailed presentation is given in Appendix B. Furthermore, a C++ and MATLAB implementation is available at `http://www.cs.cmu.edu/~mwytock/gcrf/`.

Algorithm 1 Newton Coordinate Descent for SGCRF

```
Input: Input features $X \in \mathbb{R}^{m \times n}$ and outputs $Y \in \mathbb{R}^{m \times p}$; regularization parameter $\lambda$
Output: Optimized parameters $\Lambda$, $\Theta$
Initialize: $\Lambda \leftarrow I$, $\Theta \leftarrow 0$
while (not converged) do
  1. Determine active sets $S_\Lambda$, $S_\Theta$ using (11).
  2. Find Newton update $D_\Lambda$, $D_\Theta$ by solving the following optimization using coordinate descent
       $D_\Lambda, D_\Theta \leftarrow \arg\min_{\Delta_\Lambda, \Delta_\Theta, \Delta_{\bar{S}}=0} g(\Delta_\Lambda, \Delta_\Theta) + \lambda\left(\|\Lambda + \Delta_\Lambda\|_{1,\star} + \|\Theta + \Delta_\Theta\|_1\right).$
  3. Compute a step size $\alpha$ using backtracking line search, and update
       $\Lambda \leftarrow \Lambda + \alpha D_\Lambda, \;\; \Theta \leftarrow \Theta + \alpha D_\Theta.$
end while
```

3.2. Computational speedups

In order to make the Newton method efficient numerically, there are a number of needed optimizations. Again, these mirror similar optimization presented in (Hsieh et al., 2011), but require adaptations for the CRF case. In practice, the majority of the computational work of the Newton CD method comes from computing the regularized Newton step via coordinate descent; even though coordinate descent is known to be an efficient method for solving ℓ_1 regularized problems, in our setting we have a total of $p(p+1)/2 + np$ different variables, and it would be infeasible to optimize over them all. Thus, at each iteration of the algorithm we use an active set method, and only optimize over a variable $(\Delta_\Lambda)_{ij}$ or respectively $(\Delta_\Theta)_{ij}$ if

$$\begin{aligned} |(\nabla_\Lambda f(\Lambda, \Theta))_{i,j}| > \lambda \text{ or } \Lambda_{ij} \neq 0 \\ |(\nabla_\Theta f(\Lambda, \Theta))_{i,j}| > \lambda \text{ or } \Theta_{ij} \neq 0, \end{aligned} \tag{11}$$

i.e., if the optimally conditions for that parameter are violated for the current iterate of the parameters. Because the sparsity resulting from the ℓ_1 constraint results in a relatively small active set, this provides a substantial speedup, especially when the optimal solution has high sparsity. We also keep the active set small by using warm starts; solving the optimization problem for a decreasing a sequence of the regularization parameter and initializing each successive problem with the previous optimal solution.

Second, in the coordinate descent loop, it is important to cache and incrementally update certain matrix products, such that we can evaluate subsequent coordinate updates efficiently. This requires that we maintain an explicit form of the matrix products $\Delta_\Lambda \Lambda^{-1}$ and $\Delta_\Theta \Lambda^{-1}$; crucially, when we update a single coordinate of the Δ_Θ or Δ_Λ, we only need to update a single row of these matrix products, and we can subsequently use only certain elements of these products to compute each coordinate descent step. Details are given in Appendix B.

Third, since each step of our Newton method involves solving an ℓ_1 regularized problem itself, it is important that we solve this regularized Newton step only to an accuracy that is warranted by the overall accuracy of algorithm as a whole. Although more involved approaches are possible, we simply require that the inner loop makes no more than $O(t)$ passes over the data, where t is the iteration number, a heuristic that is simple to implement and works well in practice.

Last, in cases where $n \gg m$ (which is a setting that we are crucially interested in for motivating ℓ_1 regularization), by not forming the $S_{xx} \in \mathbb{R}^{n \times n}$ matrix explicitly, we can reduce the computation for products involving $X^T X$ from $O(n^2)$ to $O(mn)$. Note that the same considerations do not apply to S_{yy}, since we need to form an invert the $p \times p$ matrix Λ to compute the gradients. Thus, the algorithm still has complexity $O(p^3)$, as in the MRF case. However, this highlights another advantage of the CRF over the MRF: when n is large, just forming a generative model over x and y jointly is prohibitively expensive. Thus, the sparse Gaussian CRF significantly improves both the performance and the computational complexity of the generative model.

4. Theoretical results

As for ℓ_1 regularized linear regression and the sparse Gaussian MRF, it is of significant interest to know when, if data is generated from a sparse underlying model, the sparse Gaussian CRF is able to recover this model with high probability. In this section, we develop theoretical results that show the sample complexity of the SGCRF grows slower than $\Omega(d^4(\log p + \log n))$ where d is the maximum degree of the output variables in the underlying graph; importantly, this term grows *logarithmically* in the input and output dimensions p and n; relative to the best known bounds for the special cases of the Gaussian MRF and linear regression, our bound has a worse dependence on d, which arises in bounding the error of the Taylor expansion. This element can likely be improved with more refined analysis, but we focus here on obtaining a bound that is logarithmic in p and n, and otherwise does not depend on on the *total* number of nonzeros in the true parameters.

The proof proceeds in the primal-dual witness (PDW) framework of Wainwright (2009) (that is, we are concerned with the setting of recovering the sparsity of the true underlying model) and our analysis mirrors much of the Gaussian MRF case (Ravikumar et al., 2011); however, as with the optimization, the additional terms in the gradient of the CRF introduce substantial added complexity, which for instance result in the worse dependence on d. We operate under the following assumptions, similar to assumptions for the Gaussian MRF and least-squares settings.

1. **True underlying model**. The data is generated according to the model
$$y|x \sim \mathcal{N}(-\Lambda^{\star -1}\Theta^{\star T}x, \Lambda^{\star -1}) \qquad (12)$$
where each row of $[\Lambda^\star \; \Theta^{\star T}]$ has at most d nonzero entries (i.e., the vertices corresponding to output variables in the graphical model of the CRF have maximum degree d). It is straightforward to generalize this to the case of sub-Gaussian noise, but we assume the Gaussian model for simplicity.

2. **Column normalization**. The columns of the input feature matrix have bounded ℓ_2 norm such that $\max_{j=1,\ldots,n} \|X_j\|_2/\sqrt{m} \le c_X$. This same assumption is used in the corresponding analysis of the ℓ_1 regularized least-squares case. Importantly, in the CRF case we make no assumptions about the distribution of x.

3. **Restricted convexity**. Letting S_i denote the nonzero indices of the ith column of $\Theta^\star$ (i.e., the edges between inputs and the ith output), we have
$$\lambda_{\min}\left(\frac{1}{m}X_{S_i}^T X_{S_i}\right) > 0 \qquad (13)$$
i.e., this term is strictly convex when restricted to the true active set. This is again a common assumption for ℓ_1 methods, but note that we do not require a restricted convexity assumption on $\Lambda^\star$ as the logdet term is already strictly convex.

4. **Mutual incoherence**. This is the most subtle of the assumptions, and one which can often be violated in practice, yet it is required for the PDW

approach. Denoting the Hessian of the objective as $H = \nabla^2_{\Lambda,\Theta} f(\Lambda, \Theta)$ and S the set of all nonzero entries of $\Lambda^\star$ and $\Theta^\star$, we require that

$$\||H_{\bar{S}S}(H_{SS})^{-1}\||_\infty \leq 1 - \alpha \qquad (14)$$

for some $\alpha > 0$ where where $\||\cdot\||_\infty$ denotes the matrix infinity norm, the maximum absolute row sum. This condition stipulates that the edges in the true active set are not too correlated with edges outside, and mirrors the same assumption for the least-squares and Gaussian MRF approaches (though of course with differences owing to the precise form of the Hessian). We give an illustrative example of this condition for simple graphs in Appendix D.

Theorem 1. *Under the above assumptions, given a sample size and regularization parameter*

$$\begin{aligned} m &\geq c_1 d^4 (1 + 8/\alpha)^2 \log(pn) \\ \lambda &\geq c_2 \alpha^{-1} \sqrt{\frac{\log(pn) + \log 4}{m}} \end{aligned} \qquad (15)$$

then with probability at least $1 - c_3 \exp(-c_4 m \lambda^2)$

1. *The solution to the ℓ_1 regularized optimization problem, $\tilde{\Lambda}$, $\tilde{\Theta}$, have nonzero entries that are a strict subset of the nonzero entries of $\Lambda^\star$, $\Theta^\star$.*
2. *The solution satisfies the elementwise bounds*

$$\begin{aligned} &\max\{\|\tilde{\Lambda} - \Lambda^\star\|_\infty, \|\tilde{\Theta} - \Theta^\star\|_\infty\} \leq \\ &c_5(1 + 8\alpha^{-1})\sqrt{\frac{\log(pn) + \log 4}{m}} \end{aligned} \qquad (16)$$

where $c_1, \ldots, c_5$ denote constant terms.

The proof of this theorem is quite lengthy and deferred to Appendix C (where we also provide an explicit definition of the constants and a lengthier definition of the assumptions). Intuitively, this theorem shows that a sample size of $m = \Omega(d^4(\log p + \log n))$ is sufficient to guarantee with high probability that solving the ℓ_1 regularized MLE recovers a subset of the true edge structure, and that the recovered parameters are close to the true parameters. Note that in many settings this is an improvement over the bound in (Yuan & Zhang, 2012), which effectively requires a sample size $\Omega(s(\log p + \log n))$ where s is the total number of edges in the graph; for graphs with a fixed low degree (such as a chain grain) s can grow linearly in p or n, whereas d remains constant, and so this represent a significant improvement—indeed, as we show in our experimental results, the empirical scaling does indeed depend only logarithmically on p, even if s increases linearly in p.[3]

5. Experimental results

Here we experimentally evaluate several aspects of the proposed model and algorithm on synthetic data and two real-world energy forecasting problems, the tasks of predicting upcoming wind power from multiple wind farms and the task of predicting upcoming electrical demand over multiple utility zones. For the latter two cases, we demonstrate state-of-the-art results. The wind prediction task is from the 2012 Global Energy Forecasting Competition (Hong, 2012), a contest recently held on Kaggle to forecast wind power; here our algorithm improves upon our own submission to this contest by 5.5% (our entry was a top-5 entry that used least-squares with the same features and was 2.5% worse than the winning entry). For load forecasting, we use real-world load data from the PJM system operator (available at `http://www.pjm.com/`) and improve upon the deployed PJM forecasts by 19%. We also highlight the performance of the algorithm relative to its theoretical bounds, and the optimization performance of our Newton method (which in all cases substantially improves upon previous methods).

5.1. Synthetic data

Exact subset recovery. Our first experiments illustrate when the model is able to exactly recover the underlying graph structure of a true model, and illustrates that the overall dependence given in our theory looks to fit the observed results. Specifically, we generate data from a chain CRF, where each output variable is connected to two others, and each input variable is connected to one output. To represent the chain we use the true parameter $\Lambda^\star$ with $\Lambda^\star_{ii} = 1$ on the diagonal, $\Lambda^\star_{ij} = 0.2$ on the super diagonal and a diagonal $\Theta^\star$ with $\Theta^\star_{ii} = 0.2$.

In Figure 2 (top), we vary m for different choices of p and observe that once m passes a certain threshold we recover the support of the true parameters with high probability—scaling the x-axis by $\log p$ demonstrates the same theoretical dependence on p as shown in our theory. Importantly, note that in this case, the total number of edges in the graph, s, increases linearly in p whereas the maximum vertex degree is fixed at $d = 3$. Thus, our bound captures the overall scaling of the model, whereas the bound of (Yuan & Zhang, 2012) would be significantly looser in this case. For the Figure 2 (bottom), we increase n by adding irrelevant features (features that are not connected to the output

[3] The bounds are not directly comparable, since Yuan & Zhang (2012) bounds only the Frobenius norm, and requires a looser restricted isometry property. Nonetheless, we see no direct way of providing a dependence on graph degree using the analysis methods in this past work, so this represents a notable improvement.

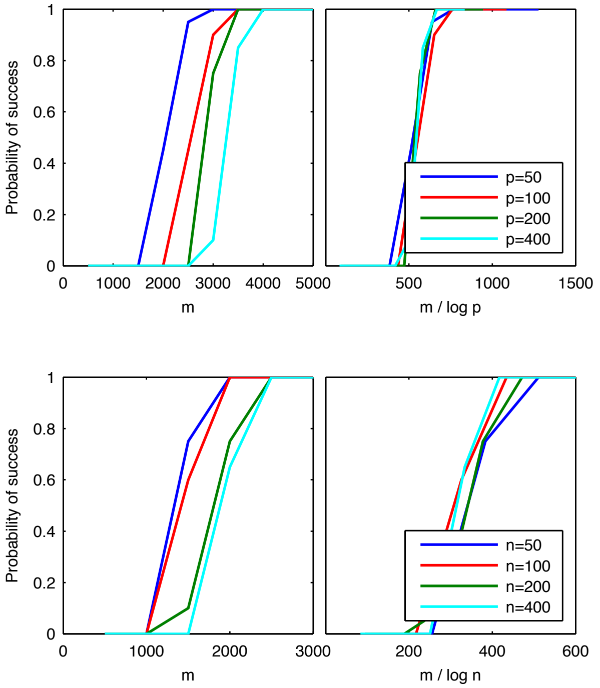

Figure 2. (top) Fraction of 20 trials in which support of the estimated parameters match that of the true parameters, increasing n, p; (bottom) adding irrelevant features, increasing n.

variables); again, we observe a logarithmic dependence on the input dimension.

Optimization performance. Because the chain CRF is a rather limited example, for the remaining synthetic examples we generate data from more complex model. In particular, follow a similar procedure as in (Yuan & Zhang, 2012), and generate Λ and Θ with $5(n + p)$ random unity entries (the rest being zero), and set the diagonal of Λ such that the condition number of Λ equals $n + p$. We sample x from a zero-mean Gaussian with full covariance, square half the entries, and then normalize the columns to have unit variance. We use this same process for the next three experiments, but vary problem size to make the experiments computationally feasible in all cases.

Figure 3 shows the suboptimality of each method in terms of the objective function $f - f^\star$ (where $f^\star$ is computed by running our Newton CD approach to numerical precision) versus execution time on a 2.4GHz Xeon processor; this problem has size $p = 1000$, $n = 4000$, and $m = 2500$. On this problem the Newton CD approach converges to high numerical precision within about 81 seconds, while FISTA and OWL-QN still don't approach this level of precision after two hours. It is also important to note that the Newton CD approach also reaches all intermediate levels of accuracy faster than the alternative approaches, so that the algorithm is preferable even if only intermediate precision is desired. Indeed, we note previous works (Sohn & Kim, 2012; Yuan & Zhang, 2012) considered maximum problem sizes of $np \approx 10^5$ due to the time required for training; since much of the appeal of ℓ_1 approaches lies precisely in the ability to use large feature sizes, this has significantly limited the applicability of the approach. We thus believe that our proposed algorithms opens the possibility of substantial new applications of this sparse Gaussian CRF model.

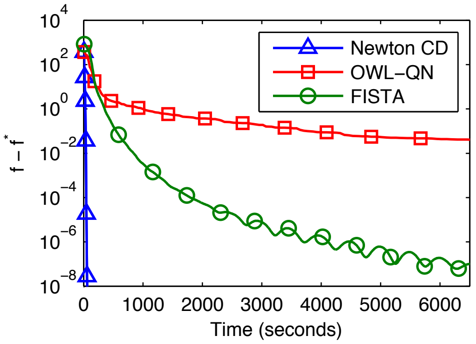

Figure 3. Suboptimality of solution versus time for Newton CD versus previously considered algorithms for sparse Gaussian CRF, OWL-QN and FISTA.

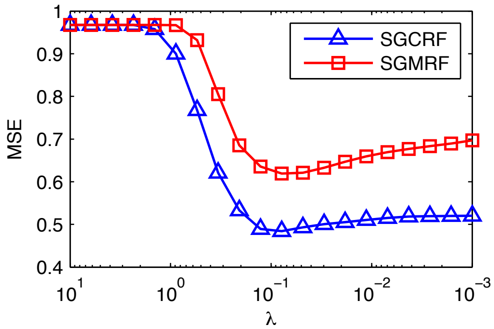

Figure 4. Generalization performance (measured by mean squared error of the predictions) for the Gaussian MRF versus CRF, with problem size $n = 200$, $p = 50$, $m = 50$.

Comparison to MRF. Our next experiment compares the discriminative CRF modeling to a generative MRF model. In particular, an alternative approach to our framework is to use a sparse Gaussian MRF to jointly model x, y as a Gaussian, then compute $y|x$. Figure 4 shows the performance of the Gaussian MRF versus CRF, measured by mean squared error in the predictions on a test set, over a variety of different λ parameters. The CRF substantially outperforms the MRF in this case, due to two main factors: 1) the x variables as generated by the above process are not Gaussian, and thus any Gaussian distribution will model them poorly; and 2) the x variables are correlated and have dense inverse covariance, making it difficult for the MRF to find a sparse solution.

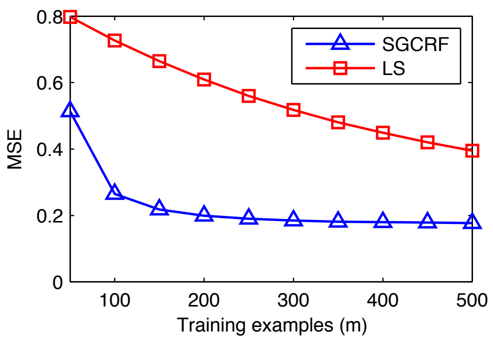

Figure 5. Generalization performance (MSE for the best λ chosen via cross-validation), for the sparse Gaussian CRF versus ℓ_2 regularized least squares. Here $n = 800$, $p = 200$.

Finally, we note again that in addition to the performance benefits, the CRF has substantial computational benefits. Modeling x and y jointly requires computing and inverting their joint covariance, which takes time $O((n+p)^3)$; in contrast, the corresponds operations for the CRF case are $O(np^3)$, which is substantially faster for even modestly large n. Indeed, for the two real-world experiments below, we were unable to successfully optimize a joint MRF using the QUIC algorithm of (Hsieh et al., 2011) (itself amongst the fastest for solving the sparse Gaussian MRF), after running the algorithm for 20 hours.

Sample size. Finally, to illustrate the benefit of ℓ_1 regularization over traditional (ℓ_2 regularized) multiple least-squares estimation, we evaluate generalization performance versus sample size, shown in Figure 5. This figure shows performance measured by mean squared error of the ℓ_1 regularized sparse Gaussian CRF versus traditional least-squares with ℓ_2 regularization; here, for each m we choose the ℓ_1 and ℓ_2 regularization parameters using a cross validation set, then evaluate the MSE on a test set. As the sample size increases, the performance of the two methods becomes similar (in the limit of infinite data with fixed n and p, they will of course be equivalent); however, as expected, for small samples sizes the ℓ_1 regularization method performs much better, being able to take advantage of the sparsity in the underlying model.

5.2. Application to energy forecasting

Wind power forecasting. We here apply the sparse Gaussian CRF model to the wind power forecasting task from the Global Energy Forecasting 2012 competition (Hong, 2012), a recent Kaggle competition for predicting upcoming wind power at seven different nearby wind farms for a time horizon of 48 hours. The input data for this problem consisted of previous power outputs for the wind farms (going as far back as the past 36 hours), and wind speed forecasts for the upcoming 48 hours. From this input we generated features that consisted of: 1) the past 8 hours of power for each wind farm, and 2) 10 RBF features placed around each forecasted wind speed, to capture non-linear dependencies on the wind speed itself. In total, this lead to $p = 336$ dimensional outputs and $n = 3417$ dimensional inputs. We heavily optimized these features for the competition, and using these features with ordinary least-squares resulted in a top-5 finish in the competition (out of 134 entrants).

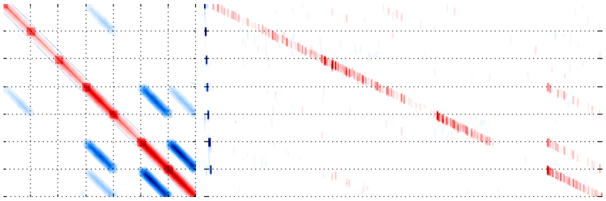

Figure 7. Sparsity patterns of estimated Λ and Θ parameters for the wind forecasting task. White denotes zero values, and a wind farms are grouped together in blocks.

Figure 6 shows the performance of the sparse Gaussian CRF on the wind forecasting task, analyzed across several dimensions. First, the figure on the left shows performance of method for varying λ; also shown in the best performance of ℓ_2 regularized least-squares. For properly chosen λ, the algorithm outperforms least-squares (using the exact same features), by 5.5%. For a domain such as wind power forecasting, where there is a limit to the possible performance (wind is an inherently stochastic phenomenon, so exact forecasts are not possible), and since the least-squares solution in this case is already using highly optimized features, this represents a substantial improvement. The difference in performance become even more pronounced for smaller sample sizes, as shown in the Figure 6 (center), which shows MSE (using λ chosen by hold out cross validation), for a variety of sample sizes. Finally, to highlight the optimization performance on real data as well, we shown in Figure 6 (right) the optimization objective versus training time for the different optimization algorithm. Again, the Newton CD algorithm vastly outperforms FISTA and OWL-QN, converging to high accuracy after 160 minutes, whereas the latter do not reach reasonable accuracy after several hours.

Finally, a significant advantage of the sparse Gaussian CRF approach is that the sparsity pattern of the resulting model can be interpreted directly as conditional dependencies between variables, and thus the sparsity pattern itself can be very informative. Figure 7 shows the sparsity patterns in Λ and Θ for the wind forecasting task; they illustrate a clear temporal and spatial dependence between the different wind farms.

Electrical demand forecasting. We further apply

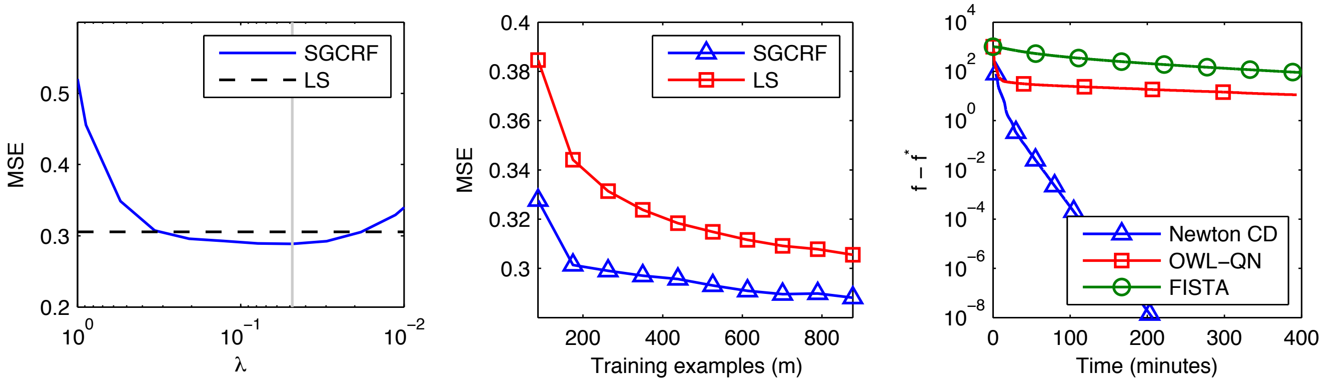

Figure 6. Performance of SGCRF on wind power forecasting showing (left) generalization performance for varying values of λ with vertical line denoting value chosen by cross-validation, (center) performance versus least-squares on different sample sizes, and (right) optimization performance of the Newton CD approach versus alternatives.

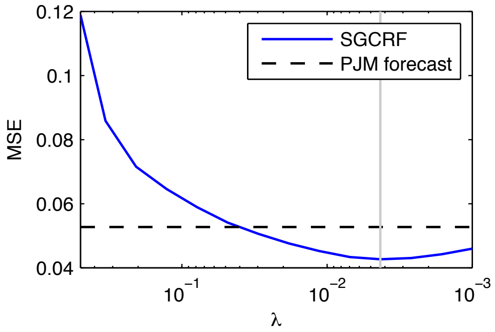

Figure 8. Generalization performance for forecasting future demand for 24 hours, compared under MSE to PJM's own forecasts with vertical line denoting λ chosen by cross-validation.

the model to the task of predicting future electrical demand for zones operated by PJM (a system operator for coordinating electricity generation and delivery for several Eastern U.S. states). In particular, the goal is to forecast upcoming electrical demand for the next 24 hours over 15 different zones in the system.

Electricity forecasting is a well-studied problem (Soliman & Al-Kandari, 2010), and PJM already employs a sophisticated forecasting system in its operation (Various, 2012) to predict a subset of the zones; rather than try to build an entirely new forecast, we *use* these previous point forecasts as input features (along with past energy consumption and time-of-day features) to predict future demand. The goal is thus to use a combination of the existing predictions to predict even more accurately, and if we can improve upon the PJM forecasts, this means that we are effectively combining existing information to ultimately deliver a better prediction. For this problem, the dimension of input and output are $p = 350$ and $n = 860$. We present these results here more briefly, but the key performance element we want to highlight is in Figure 8; the figure shows that by jointly predicting over all the zones, we are able to improve substantially upon PJM's already state-of-the-art point forecasts.

6. Conclusion and discussion

The sparse Gaussian conditional random field enjoys many benefits of existing methods for learning high-dimensional Gaussian graphical models; we believe that the advances put forward in this paper make the model significantly more practical for large-scale problems, and also significantly advance our theoretical understanding of the method. Furthermore, the empirical results presented here on wind power and demand forecasting are of substantial practical interest, and the improvements presented here have the potential for notable effects on power system efficiency.

Two future directions seem particularly promising. First, it would be worthwhile to use regret-based approaches to develop alternate convergence rates under weaker assumptions than those we use. Although exact feature selection is not possible even for the least-squares case when inputs are very highly correlated, it is nonetheless possible to obtain regret bounds that bound the *loss* versus that of the true model, e.g. (Bartlett et al., 2012); such directions are likely to be of substantial interest here, since we do expect to often be in situations where input features are correlated. Second, from an application standpoint in energy systems in particular, there are a huge number of forecasting problems that share similar properties as wind power and demand; of crucial importance, however, is developing control algorithms that can exploit these probabilistic forecasts in the planning stage. Developing such algorithms will allow high-dimensional graphical models such as the Gaussian CRF to have an immediate impact on these globally important domains.

References

Andrew, Galen and Gao, Jianfeng. Scalable training of l 1-regularized log-linear models. In *Proceedings of the 24th international conference on Machine learning*, pp. 33–40. ACM, 2007.

Banerjee, O., Ghaoui, L. El, and d'Aspremont, A. Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data, 2008.

Bartlett, Peter L, Mendelson, Shahar, and Neeman, Joseph. ℓ_1-regularized linear regression: Persistence and oracle inequalities. *Probability theory and related fields*, pp. 1–32, 2012.

Beck, Amir and Teboulle, Marc. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2 (1):183–202, 2009.

Duchi, J. C., Gould, S., and Koller, D. Projected subgradient methods for learning sparse Gaussians. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2008.

Friedman, J., Hastie, T., and Tibshirani, R. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

Hong, T. Global energy forecasting competition, 2012. URL http://www.gefcom.org.

Hsieh, C.-J., Sustik, M. A., Dhillon, I. S., and Ravikumar, P. Sparse inverse covariace matrix estimation using quadratic approximation. In *Neural Information Processing Systems*, 2011.

Lu, Z. Smooth optimization approaches for sparse inverse covariance selection. *SIAM Journal on Optimization*, 19(4):1807–1827, 2009.

Ng, A. Y. and Jordan, M. I. On discriminative vs. generative classifiers: a comparison of logistic regression and naive Bayes. 2002.

Olsen, Peder A, Oztoprak, Figen, Nocedal, Jorge, and Rennie, Stephen J. Newton-like methods for sparse inverse covariance estimation. *Optimization Online*, 2012.

Ravikumar, Pradeep, Wainwright, Martin J, Raskutti, Garvesh, and Yu, Bin. High-dimensional covariance estimation by minimizing 1-penalized log-determinant divergence. *Electronic Journal of Statistics*, 5:935–980, 2011.

Scheinberg, K., Ma, S., and Goldfarb, D. Sparse inverse covariance selection via alternating linearization methods. In *Neural Information Processing Systems*, 2010.

Sohn, Kyung-Ah and Kim, Seyoung. Joint estimation of structured sparsity and output structure in multiple-output regression via inverse-covariance regularization. In *Proceedings of the Conference on Artificial Intelligence and Statistics*, 2012.

Soliman, S. A. and Al-Kandari, A. M. *Electrical Load Forecasting: Modeling and Model Construction*. Elsevier, 2010.

Sutton, C. and McCallum, A. An introduction to conditional random fields. *Foundations and Trends in Machine Learning*, 4(4):267–373, 2012.

Tseng, Paul and Yun, Sangwoon. A coordinate gradient descent method for nonsmooth separable minimization. *Mathematical Programming*, 117(1):387–423, 2009.

Various. *PJM Manual 19: Load Forecasting and Analysis*. PJM, 2012. Available at: http://www.pjm.com/planning/resource-adequacy-planning/~/media/documents/manuals/m19.ashx.

Wainwright, M.J. Sharp thresholds for high-dimensional and noisy sparsity recovery using ℓ_1-constrained quadratic programming (Lasso). *IEEE Transactions on Information Theory*, 55(5):2183–2202, 2009.

Wytock, Matt and Kolter, J. Zico. Sparse conditional gaussian random fields. In *NIPS Workshop on Log Linear Models*, 2012.

Yuan, Xiao-Tong and Zhang, Tong. Partial gaussian graphical model estimation. *CoRR*, abs/1209.6419, 2012.

Appendices to "Sparse Gaussian Conditional Random Fields: Algorithms, Theory, and Application to Energy Forecasting"

A. Derivation of Gradient, Hessian, and Differentials

Here we derive analytic expressions for the gradient, Hessian, and various differentials of the log likelihood function. Recall that the log-likelihood is given by

$$f(\Lambda, \Theta) = -\log|\Lambda| + \mathrm{tr}\Lambda S_{yy} + 2\mathrm{tr}\Theta^T S_{xy} + \mathrm{tr}\Lambda^{-1}\Theta^T S_{xx}\Theta \tag{1}$$

We adopt the differential matrix calculus notation from (Magnus & Neudecker, 1988) where for a matrix $A \in \mathbb{R}^{m\times n}$ and a function $f : \mathbb{R}^{m\times n} \to \mathbb{R}$, $d^k f(A;U) : \mathbb{R}^{m\times n} \times \mathbb{R}^{m\times n} \to \mathbb{R}$ denotes the k-th differential of the function f evaluated at U. For example, the first differential can easily be expressed in terms of the gradient

$$df(A;U) = \mathrm{tr}\nabla_A f(A)^T U. \tag{2}$$

Other derivatives for functions of matrices (i.e., Hessians or higher order terms) are cumbersome to represent directly, but the differentials themselves can typically be expressed compactly; indeed, it is often simplest to first derive these differentials and then use them to determine analytical expressions for the Hessians and higher order derivatives. Furthermore, the Taylor expansion of a function can be represented directly in terms of its differentials; for instance the second order approximation is given by

$$f(A+\Delta) \approx f(A) + df(A;\Delta) + \frac{1}{2}d^2 f(A;\Delta) \equiv f(A) + \mathrm{vec}(\nabla_A f(A))^T \mathrm{vec}(\Delta) + \frac{1}{2}\mathrm{vec}(\Delta)^T(\nabla_A^2 f(A))\,\mathrm{vec}(\Delta) \tag{3}$$

where vec denotes the vectorization of a matrix (forming a vector by concatenating the columns), and $\nabla_A^2 f(A)$ denotes the Hessian.

Using standard rules of differential calculus, we can compute the first and second order differentials of the log-likelihood $f(\Lambda, \Theta)$,

$$df(\Lambda,\Theta;U,V) = \mathrm{tr}S_{yy}U + 2\mathrm{tr}S_{yx}V - \mathrm{tr}\Lambda^{-1}U + 2\mathrm{tr}\Lambda^{-1}\Theta^T S_{xx}V - \mathrm{tr}\Lambda^{-1}\Theta^T S_{xx}\Theta\Lambda^{-1}U \tag{4}$$

and from this expression we can easily determine the relevant gradients

$$\begin{aligned}\nabla_\Lambda f(\Lambda,\Theta) &= S_{yy} - \Lambda^{-1} - \Lambda^{-1}\Theta^T S_{xx}\Theta\Lambda^{-1}\\ \nabla_\Theta f(\Lambda,\Theta) &= 2S_{xy} + 2S_{xx}\Theta\Lambda^{-1}.\end{aligned} \tag{5}$$

Similarly, we can differentiate again to find the second differential

$$d^2 f(\Lambda,\Theta;U,V) = 2\mathrm{tr}\Lambda^{-1}U\Lambda^{-1}\Theta^T S_{xx}\Theta\Lambda^{-1}U + \mathrm{tr}\Lambda^{-1}U\Lambda^{-1}U + 2\mathrm{tr}\Lambda^{-1}V^T S_{xx}V - 4\mathrm{tr}\Lambda^{-1}U\Lambda^{-1}\Theta^T S_{xx}V. \tag{6}$$

Combining the first and second differential gives the full second order Taylor expansion shown in the paper. It also lets us determine the Hessian itself, which we use in the incoherence condition for the theoretical results

$$\nabla^2_{\Lambda,\Theta} f(\Lambda,\Theta) = \begin{bmatrix}\Lambda^{-1}\otimes(\Lambda^{-1} + 2\Lambda^{-1}\Theta^T S_{xx}\Theta\Lambda^{-1}) & -2\Lambda^{-1}\otimes\Lambda^{-1}\Theta^T S_{xx}\\ -2\Lambda^{-1}\otimes S_{xx}\Theta\Lambda^{-1} & 2\Lambda^{-1}\otimes S_{xx}\end{bmatrix} \tag{7}$$

B. Detailed Description of Newton Coordinate Descent Method

We present a detailed description and full pseudo-code for the Newton coordinate descent algorithm. The derivation mirrors that in (Hsieh et al., 2011). The complete method is shown in Algorithm 1, with the coordinate

Algorithm 1 Newton Coordinate Descent for SGCRF

```
  Input: Input features $X \in \mathbb{R}^{m\times n}$ and outputs $Y \in \mathbb{R}^{m\times p}$; regularization parameter $\lambda$; algorithm parameters $\epsilon$, $\sigma$, $\alpha$, $\beta$.
  Output: Optimized parameters $\Lambda$, $\Theta$
  Initialize: $\Lambda \leftarrow I$, $\Theta \leftarrow 0$, $\Sigma \leftarrow \Lambda^{-1}$
  while (not converged) do
    1. Compute the gradient, determine active sets $S_\Lambda$, $S_\Theta$ using (14), and check for convergence.
    2. Find regularized Newton direction $D_\Lambda$, $D_\Theta$ using Algorithm 2.
    3. Initialize $\alpha \leftarrow 1$ and compute

       $$\mu \leftarrow (\mathrm{tr}\nabla_\Lambda f(\Lambda,\Theta)^T\Delta_\Lambda + \mathrm{tr}\nabla_\Theta f(\Lambda,\Theta)^T\Delta_\Theta + \|\Lambda+\Delta_\Lambda\|_{1\star} + \|\Theta+\Delta_\Theta\|_1).$$

    while (insufficient descent) do
      1. Compute the Cholesky decomposition $LL^T = \Lambda + \alpha D_\Lambda$, continuing if not positive definite
      2. Check descent $f(\Lambda+\alpha D_\Lambda, \Theta+\alpha D_\Theta) < f(\Lambda,\Theta) + \alpha\sigma\mu$ and break if satisfied
      3. $\alpha \leftarrow \beta\alpha$
    end while
  end while
```

descent inner loop for computing the approximation to the Newton direction given in Algorithm 2. This process repeats until the solution converges to a within a specified tolerance, checked using the KKT conditions.

Next, we derive the coordinatewise updates for the inner loop and highlight the key optimizations that are used in order to achieve fast performance.

B.1. Coordinate descent updates for the Newton approximation

To begin, note that for a fixed Λ and Θ, the regularized Newton direction is given by the solution to the second-order Taylor expansion which for our problem has the form

$$\begin{aligned} h(\Delta_\Lambda, \Delta_\Theta) = \; & \mathrm{tr}S_{yy}\Delta_\Lambda + 2\mathrm{tr}S_{yx}\Delta_\Theta - \mathrm{tr}\Lambda^{-1}\Delta_\Lambda \; + 2\mathrm{tr}\Lambda^{-1}\Theta^T S_{xx}\Delta_\Theta - \\ & \mathrm{tr}\Lambda^{-1}\Theta^T S_{xx}\Theta\Lambda^{-1}\Delta_\Lambda + \mathrm{tr}\Lambda^{-1}\Delta_\Lambda\Lambda^{-1}\Theta^T S_{xx}\Theta\Lambda^{-1}\Delta_\Lambda - \frac{1}{2}\mathrm{tr}\Lambda^{-1}\Delta_\Lambda\Lambda^{-1}\Delta_\Lambda + \\ & \lambda\left(||\Lambda+\Delta_\Lambda\|_{1,\star} + \|\Theta+\Delta_\Theta\|_1\right) \end{aligned} \tag{8}$$

We split the updates into three cases. First, we consider optimizing over a diagonal element of D_Λ by finding $\mu = \arg\min_\mu h(\Delta_\Lambda + \mu e_i e_i^T, \Delta_\Theta)$ which has the explicit form

$$\begin{aligned} \underset{\mu}{\text{minimize}} \;\; & \frac{1}{2}\mu^2\left[\Sigma_{ii}^2 + 2\Sigma_{ii}\Psi_{ii}\right] + \mu\left[-\Sigma_{ii} + (S_{yy})_{ii} - \Psi_{ii} + (\Sigma U\Sigma)_{ii} - 2(\Sigma\Theta^T S_{xx}V\Sigma)_{ii} + 2(\Psi U\Sigma)_{ii}\right] + \\ & \lambda|\Lambda_{ii} + \mu| \end{aligned} \tag{9}$$

where $\Sigma = \Lambda^{-1}$ and $\Psi = \Sigma\Theta^T S_{xx}\Theta\Sigma$.

Next, note that for two symmetric matrices A, B, not necessarily equal, the symmetric update is given by

$$\begin{aligned} & \arg\min_\mu \;\; \mathrm{tr}A(U + \mu(e_i e_j^T + e_j e_i^T))B(U + \mu(e_i e_j^T + e_j e_i^T)) \\ = & \arg\min_\mu \;\; \mu^2\mathrm{tr}A(e_i e_j^T + e_j e_i^T)B(e_i e_j^T + e_j e_i^T) + \mu\mathrm{tr}AUB(e_i e_j^T + e_j e_i^T) + \mu\mathrm{tr}A(e_i e_j^T + e_j e_i^T)BU \\ = & \arg\min_\mu \;\; \mu^2(A_{ii}B_{jj} + 2A_{ij}B_{ij} + A_{jj}B_{ii}) + 2\mu((AUB)_{ij} + (AUB)_{ji}) \end{aligned} \tag{10}$$

Applying this equivalence twice, once with $A = B = \Sigma$ and again with $A = \Sigma$, $B = \Psi$ the the symmetric update

Algorithm 2 Coordinate descent inner loop

```
Input: $S$ empirical covariance, $\lambda$ regularization parameter, $S_\Lambda, S_\Theta$ active sets and $\Lambda, \Theta$ current estimates
Output: Approximate regularized Newton direction $D_\Lambda$, $D_\Theta$
Initialize: $D_\Lambda \leftarrow 0$, $D_\Theta \leftarrow 0, U \leftarrow 0, V \leftarrow 0$
while (not converged) do
  for coordinate $(i,j)$ in $S_\Lambda$ do
    1. Find $\mu$ by solving (9) or (11), using $U$ and $V$ for efficiency.
    2. Symmetrically update $D_\Lambda$ and two rows of $U$

          $(D_\Lambda)_{ij}, (D_\Lambda)_{ij} \leftarrow (D_\Lambda)_{ij} + \mu$
                                  $U_i \leftarrow U_i + \mu\Sigma_j$
                                  $U_j \leftarrow U_j + \mu\Sigma_i$

    where $\Sigma_i$ denotes the ith row of $\Lambda^{-1}$.
  end for
  for coordinate $(i,j)$ in $S_\Theta$ do
    1. Find $\mu$ by solving (12), using $U$ and $V$ for efficiency.
    2. Update $D_\Theta$ and one row of $V$

          $(D_\Theta)_{ij} \leftarrow (D_\Theta)_{ij} + \mu$
                  $V_i \leftarrow V_i + \mu\Sigma_j$

  end for
end while
```

for an off-diagonal element of matrix D_Λ, $\mu = \arg\min_\mu h(\Delta_\Lambda + \mu(e_i e_j^T + e_j e_i^T), \Delta_\Theta)$ is given by

$$\begin{aligned}\underset{\mu}{\text{minimize}}\ \ & \mu^2\left[\Sigma_{ij}^2 + \Sigma_{ii}\Sigma_{jj} + \Sigma_{ii}\Psi_{jj} + 2\Sigma_{ij}\Psi_{ij} + \Sigma_{jj}\Psi_{ii}\right] + \\ & 2\mu\left[-\Sigma_{ij} + (S_{yy})_{ij} - \Psi_{ij} + (\Sigma U \Sigma)_{ij} - \Phi_{ij} - \Phi_{ji} + (\Psi U \Sigma)_{ij} + (\Psi U \Sigma)_{ji}\right] + \\ & 2\lambda|\Lambda_{ij} + U_{ij} + \mu|\end{aligned} \tag{11}$$

where $\Phi = \Sigma\Theta^T S_{xx} V \Sigma$. Finally, we consider optimizing over an element of D_Θ

$$\begin{aligned}\underset{\mu}{\text{minimize}}\ \ & \mu^2\left[\Sigma_{jj}(S_{xx})_{ii}\right] + \mu\left[2(S_{xy})_{ij} + 2(S_{xx}\Theta\Sigma)_{ij} + 2(S_{xx}V\Sigma)_{ij} - 2(S_{xx}\Theta\Sigma U\Sigma)_{ij}\right] + \\ & \lambda|\Theta_{ij} + V_{ij} + \mu|\end{aligned} \tag{12}$$

Each equation has a quadratic form and thus can be solved in closed form. The second two have an ℓ_1 penalty and the form $\min_\mu \frac{1}{2}a\mu^2 + b\mu + \lambda|c + \mu|$ which has the solution

$$\mu = -c + S_{\lambda/a}\left(c - \frac{b}{a}\right) \tag{13}$$

B.2. Optimizations

As in the case of the MRF (Hsieh et al., 2011), there are several modifications to a naive solution that significantly reduce the running time of the algorithm.

First, consider the matrix products involved in the coordinatewise updates above. A naive implementation of the coordinate descent algorithm would require $O((n+p)^2)$ operations even though the majority of the elements are unchanged from one iteration. However, by caching products of the static matrices and maintaining a factorized form of the products involving Δ_Λ and Δ_Θ, specifically $U = \Delta_\Lambda\Sigma$, $V = \Delta_\Theta\Sigma$, we reduce this to $O((n+p))$. As a consequence, at each iteration of the loop we must update the rows of U and V corresponding to the coordinates of Δ_Λ and Δ_Θ.

Next, we describe how we drastically reduce the coordinate descent active set. At each iteration of the outer loop, we fix the active set using the current nonzero coordinates and the KKT conditions of the objective function. We include a coordinate of Λ, respectively Θ, if

$$\begin{aligned} |(\nabla_\Lambda f(\Lambda, \Theta))_{i,j}| &> \lambda \text{ or } \Lambda_{ij} \neq 0 \\ |(\nabla_\Theta f(\Lambda, \Theta))_{i,j}| &> \lambda \text{ or } \Theta_{ij} \neq 0. \end{aligned} \tag{14}$$

Since the size of this active set is determined by the number of nonzero elements in the parameters, for sparse solutions the speed up is very significant. Note that although we fix the active set before beginning coordinate descent, as in the MRF case (Hsieh et al., 2011), we still have convergence guarantees for the overall algorithm.

Finally, note that when taking a step we must ensure sufficient descent and that the Λ parameter remains in the semidefinite cone. We accomplish this using the Cholesky decomposition, which is also used for efficiently computing Λ^{-1}.

C. Theoretical Analysis

We will make the following assumptions about the input and output variables X and Y. These mirror similar assumptions in (Wainwright, 2009) and (Ravikumar et al., 2011), and we will discuss the precise differences.

First, the analysis here proceeds on the assumption that there is a true underlying model generating the test data, of the prescribed form (i.e., the data is generated according to a sparse Gaussian CRF). It is trivial to extend this analysis to the case of sub-Gaussian noise, but we simply assume Gaussian noise for simplicity of presentation

Assumption 1. ***Underlying model*** *The data is generated according to*

$$y|x \sim \mathcal{N}(-\Lambda^{\star -1}\Theta^{\star T}x, \Lambda^{\star -1}). \tag{15}$$

where each row of $[\Lambda^\star \; \Theta^{\star T}]$ has at most d nonzero entries (i.e., the vertices corresponding to output variables in the graphical model of the CRF have maximum degree d).

For simplicity, we will also denote $\Sigma^\star = \Lambda^{\star -1}$.

Our second assumption is a restricted convexity requirement, which ensures that the optimization problem restricted to the active set is unique. This is a common assumption for ℓ_1 approaches (the same condition appears in the least-squares analysis of (Wainwright, 2009)), and the only extension here is that we require this to hold for each output variable.

Assumption 2. ***Restricted convexity*** *For each output i, let S_i denote $\{j : \Theta_{ji} \neq 0\}$ (i.e., S_i is the “active set” of edges directly connecting an input to y_i), we have that*

$$\lambda_{\min}\left(\frac{1}{m} X_{S_i}^T X_{S_i}\right) > 0. \tag{16}$$

The next assumption is more subtle (and quite strict in practice), but is again typical for exact subset selection proofs for ℓ_1 approaches. Namely, we require a mutual incoherence assumption, which effectively ensures that the connections in the CRF that correspond to the “true” edges do not correlate too much with edges that are not the support set.

Assumption 3. ***Mutual incoherence*** *Let S denote the active set of all variables in vector form*

$$S = \begin{bmatrix} \operatorname{vec}(\operatorname{supp}\{\Lambda^\star\}) \\ \operatorname{vec}(\operatorname{supp}\{\Theta^\star\}) \end{bmatrix} \tag{17}$$

where supp *denotes the support function (the indicator of whether an element is nonzero), and let $\bar{S}$ denote its complement. Then for $H = \nabla^2_{\Lambda,\Theta} f(\Lambda, \Theta)$ defined above*

$$|||H_{\bar{S}S}(H_{SS})^{-1}|||_\infty \leq 1 - \alpha \tag{18}$$

for some $\alpha > 1$, where $|||\cdot|||_\infty$ denotes the matrix infinity norm, the maximum absolute row sum.

Our first lemma shows that the gradients $\nabla_\Lambda f(\Lambda^\star, \Theta^\star)$ and $\nabla_\Theta f(\Lambda^\star, \Theta^\star)$ (the gradients evaluated at the true parameters) are small (in infinity norm) with high probability given samples on the order of $m = \Omega(\log n + \log p)$. The proof (shown below) follows from a standard bound on Gaussian random variables, and from Lemma 1 in (Ravikumar et al., 2011).

Lemma 1. *Given data generated by the model in Assumption 1 we have that*

$$P\left(\|\nabla_\Theta f(\Lambda^\star, \Theta^\star)\|_\infty > \epsilon\right) \leq 2np \exp\left\{-\frac{m\epsilon^2}{8c_{\sigma^\star}^2 c_X^2}\right\} \tag{19}$$

where $c_{\sigma^\star} = \max_i \Sigma^\star_{ii}$, and $c_X = \max_{j=1,\ldots,n} \|X_j\|_2/\sqrt{m}$; the maximum normalized ℓ_2 norm over columns of X. Furthermore,

$$P\left(\|\nabla_\Lambda f(\Lambda^\star, \Theta^\star)\|_\infty > \epsilon\right) \leq 4p^2 \exp\left\{-\frac{m\epsilon^2}{3200c_{\sigma^\star}^2}\right\} \tag{20}$$

for $0 < \delta < 40c_{\sigma^\star}$.

The next lemma is a generic primal-dual witness approach, mirroring exactly the derivation in (Wainwright, 2009), but presented in a generic form. For the presentation here, we will use a generic optimization problem minimize $f(\theta) + \lambda\|\theta\|_1$, though we will apply this specifically to our CRF problem momentarily. Intuitively, the lemma states conditions for which optimizing over the known support set is equivalent to optimizing with the ℓ_1 penalty.

Lemma 2. *Consider some sparse $\theta^\star$ with $S = \operatorname{supp}(\theta^\star)$, and consider the two optimization problems*

$$\begin{aligned}\hat{\theta} &= \arg\min_{\theta} f(\theta) + \lambda\|\theta\|_1 \\ \tilde{\theta} &= \arg\min_{\theta, \theta_{\bar{S}}=0} f(\theta) + \lambda\|\theta\|_1.\end{aligned} \tag{21}$$

Define $\Delta = \tilde{\theta} - \theta^\star$, and $R(\Delta) = -\nabla_\theta f(\tilde{\theta}) + \nabla_\theta f(\theta^\star) + \nabla_\theta^2 f(\theta^\star)\Delta$. Then if the following conditions hold

1. *The solution $\tilde{\theta}$ is unique.*
2. *Mutual incoherence holds, i.e., $\|\!|(\nabla_\theta^2 f(\theta^\star))_{\bar{S}S}(\nabla_\theta^2 f(\theta^\star))_{SS}^{-1}\|\!|_\infty \leq 1 - \alpha$*
3. $\max\{\|\nabla_\theta f(\theta^\star)\|_\infty, \|R(\Delta)\|_\infty\} \leq \lambda\alpha/8$

then the ℓ_1 solution recovers the restricted solution, $\tilde{\theta} = \hat{\theta}$.

In our setting, the Δ and $R(\Delta)$ are themselves matrices and thus slightly more complex. Thus, for subsequent lemmas, we define the following terms that we use to quantity the error of the second order Taylor expansions of our particular log-likelihood, evaluated at the true parameters. For any Λ, Θ, we define $\Delta_\Lambda = \Lambda - \Lambda^\star$ and $\Delta_\Theta = \Theta - \Theta^\star$ and

$$\Delta \equiv \begin{bmatrix} \Lambda \\ \Theta \end{bmatrix}. \tag{22}$$

Define

$$\begin{aligned} R_\Lambda(\Delta_\Lambda, \Delta_\Theta) &= \nabla_\Lambda f(\Lambda^\star, \Theta^\star) - \nabla_\Lambda f(\Lambda^\star + \Delta_\Lambda, \Theta^\star + \Delta_\Theta) + d(\nabla_\Lambda f(\Lambda^\star, \Theta^\star); \Delta_\Lambda, \Delta_\Theta) \\ R_\Theta(\Delta_\Lambda, \Delta_\Theta) &= \nabla_\Theta f(\Lambda^\star, \Theta^\star) - \nabla_\Theta f(\Lambda^\star + \Delta_\Lambda, \Theta^\star + \Delta_\Theta) + d(\nabla_\Theta f(\Lambda^\star, \Theta^\star); \Delta_\Lambda, \Delta_\Theta), \end{aligned} \tag{23}$$

which are the residuals of the first order Taylor expansion of the gradient (i.e., the errors in the second order Taylor expansion of the function itself), and

$$R(\Delta) = \begin{bmatrix} R_\Lambda(\Delta_\Lambda, \Delta_\Theta) \\ R_\Theta(\Delta_\Lambda, \Delta_\Theta) \end{bmatrix}. \tag{24}$$

The next lemma bounds the residual $\|R(\Delta)\|_\infty$ in terms of the distance from the true parameters, $\|\Delta\|_\infty$.

Lemma 3. *Under the definitions above, if*

$$\|\Delta\|_\infty \leq \frac{1}{d} \min\left\{\frac{1}{3c_{\Sigma^\star}}, \frac{c_{\Theta^\star}}{2}\right\} \tag{25}$$

then

$$\|R(\Delta)\|_\infty \leq 206 c_{\Sigma^\star}^4 c_{\Theta^\star}^2 c_X^2 d^2 \|\Delta\|_\infty^2 \tag{26}$$

where $c_{\Sigma^\star} = \max_{i,j} \Sigma^\star_{ij}$ *and* $c_{\Theta^\star} = \max_{i,j} \Theta^\star_{ij}$.

Finally, we show that when the gradient evaluated at the true model are sufficiently small, then $\|\Delta\|_\infty$ itself is small.

Lemma 4. *Under the model above, suppose that*

$$\max\{\|\nabla_\Lambda f(\Lambda^\star, \Theta^\star)\|_\infty, \|\nabla_\Theta f(\Lambda^\star, \Theta^\star)\|_\infty\} \leq \frac{1}{2c_{H^\star}} \left[\min\left\{\frac{1}{3c_{\Sigma^\star} d}, \frac{1}{412 c_{\Sigma^\star}^4 c_{\Theta^\star}^2 c_X^2 d^2}\right\} - \lambda\right]. \tag{27}$$

Then

$$\|\Delta\|_\infty \leq 2c_{H^\star}(\max\{\|\nabla_\Lambda f(\Lambda^\star, \Theta^\star)\|_\infty, \|\nabla_\Theta f(\Lambda^\star, \Theta^\star)\|_\infty\} + \lambda) \tag{28}$$

where $c_{H^\star} = \max_{i,j} H^\star_{ij}$*, the maximum element of the Hessian evaluated at the true parameters.*

These elements allow us to prove the desired theorem.

Theorem 1. *Using assumptions 1-3 above, suppose we have sample size*

$$m \geq 412^2 C^2 d^4 (1 + 8/\alpha)^2 \log(pn) \tag{29}$$

where $C = \max\{3c_{\Sigma^\star}, c_{\Theta^\star}^{-1}, c_{\Sigma^\star}^4 c_{\Sigma^\star}^2 c_X^2\}$ *and choose* λ *as*

$$\lambda \geq (8/\alpha) c_{\sigma^\star} c_X \sqrt{3200} \sqrt{\frac{\log(pn) + \log 4}{m}} \tag{30}$$

then with probability greater than $1 - c_1 \exp(-c_2 m \lambda^2)$ *we have*

1. *The solution to the* ℓ_1 *regularized optimization problem,* $\tilde{\Lambda}$, $\tilde{\Theta}$*, has nonzero entries that are a strict subset of the nonzero entries of* $\Lambda^\star$, $\Theta^\star$
2. *The solution satisfies the elementwise bounds*

$$\max\{\|\tilde{\Lambda} - \Lambda^\star\|_\infty, \|\tilde{\Theta} - \Theta^\star\|_\infty\} \leq 2(1 + 8\alpha^{-1}) c_{H^\star} c_{\sigma^\star} c_X \sqrt{3200} \sqrt{\frac{\log(pn) + \log 4}{m}} \tag{31}$$

Proof. Let

$$\delta = c_{\sigma^\star} c_X \sqrt{3200} \sqrt{\frac{\log(pn) + \log 4}{m}}. \tag{32}$$

Then by Lemma 1 and the minimum bound on m we have that

$$\max\{\|\nabla_\Theta f(\Lambda^\star, \Theta^\star)\|_\infty, \|\nabla_\Lambda f(\Lambda^\star, \Theta^\star)\|_\infty\} \leq \delta \tag{33}$$

with probability greater than $1 - c_1 \exp(-c_2 m \lambda^2)$; we proceed with the proof conditioned on this event. Next, note by our choice of λ we have that $\delta \leq \alpha\lambda/8$ and thus the first half of the third condition for Lemma 2 holds. It remains to show that $R(\Delta) \leq \alpha\lambda/8$. By our minimum bound on m and our choice of λ we have that

$$\left(1 + \frac{8}{\alpha}\right) \delta \leq \frac{1}{2c_{H^\star}} \min\left\{\frac{1}{3c_{\Sigma^\star} d}, \frac{1}{412 c_{\Sigma^\star}^4 c_{\Theta^\star}^2 c_X^2 d^2}\right\} \tag{34}$$

and thus Lemma 4 applies, which gives

$$\|\Delta\|_\infty \leq 2c_{H^\star} \left(1 + \frac{8}{a}\right) \delta. \tag{35}$$

Therefore, the assumption of $\|\Delta\|_\infty \leq \frac{1}{d}\min\{\frac{1}{3c_{\Sigma^\star}}, \frac{c_{\Theta^\star}}{2}\}$ holds and we apply Lemma 3 to establish

$$\begin{aligned}\|R(\Delta)\|_\infty &\leq 206c^4_{\Sigma^\star}c^2_{\Theta^\star}c^2_X d^2\|\Delta\|^2_\infty \\ &\leq 824c^4_{\Sigma^\star}c^2_{\Theta^\star}c^2_X d^2 c_{H^\star}\left(1+\frac{8}{a}\right)^2\delta^2 \\ &\leq \left[824c^4_{\Sigma^\star}c^2_{\Theta^\star}c^2_X d^2 c_{H^\star}\left(1+\frac{8}{a}\right)^2\delta\right]\frac{\alpha\lambda}{8} \\ &\leq \frac{\alpha\lambda}{8}.\end{aligned} \tag{36}$$

Finally, note that our Assumption 2 implies that the solution $(\tilde{\Lambda}, \tilde{\Theta})$ is unique and thus combined with the above derivation and Assumption 3, we the conditions for Lemma 2 and thus we conclude that $(\tilde{\Lambda}, \tilde{\Theta}) = (\hat{\Lambda}, \hat{\Theta})$ and the thus claim 1 and 2 are satisfied. □

C.1. Proofs of Lemmas

Proof. (of Lemma 1) Let X be given and assuming that Y is generated according to our model. We first consider $P\left(\|\nabla_\Theta f(\Lambda^\star, \Theta^\star)\|_\infty > \epsilon\right)$; as shown in Appendix A, we have

$$\nabla_\Theta f(\Lambda, \Theta) = 2S_{xy} + 2S_{xx}\Theta\Lambda^{-1}. \tag{37}$$

Writing $\beta^* = -\Theta^*\Lambda^{*-1}$ and $\Sigma^* = \Lambda^{*-1}$, we have $Y = X\beta^* + Z$ where $Z \in \mathbb{R}^{m\times p}$ has rows $Z_i \sim \mathcal{N}(0, \Sigma^*)$, and thus

$$2S_{xy} + 2S_{xx}\Theta^*\Lambda^{*-1} = \frac{2}{m}(X^TY - X^TX\beta^*) = \frac{2}{m}X^TZ. \tag{38}$$

By our assumptions that $\|X_j\|_2/\sqrt{n} < c_X$ for all columns of X and the maximum diagonal entry $\Sigma^\star$ is $c^2_{\sigma^\star}$, we have

$$\text{Var}\left(\frac{2}{m}X_i^T Z_j\right) \leq \frac{4c^2_{\sigma^\star}c^2_X}{m} \tag{39}$$

for any columns X_i and Z_j. Therefore by the union bound and Gaussian tail probability we have

$$P\left(\|\frac{1}{m}X^TZ\|_\infty > \epsilon\right) \leq 2np\exp\left\{-\frac{m\epsilon^2}{8c^2_{\sigma^*}c^2_X}\right\} \tag{40}$$

Next, we consider $P\left(\|\nabla_\Lambda f(\Lambda^\star, \Theta^\star)\|_\infty > \epsilon\right)$ and again from Appendix A, we have

$$\nabla_\Lambda f(\Lambda, \Theta) = S_{yy} - \Lambda^{-1} - \Lambda^{-1}\Theta^T S_{xx}\Theta\Lambda^{-1} \tag{41}$$

which we can rewrite as

$$S_{yy} - \Lambda^{*-1} - \Lambda^{*-1}\Theta^{*T}S_{xx}\Theta^*\Lambda^{*-1} = \frac{1}{m}Z^TZ - \Sigma^*. \tag{42}$$

Now we can apply Lemma 1 in (Ravikumar et al., 2011) and arrive at the desired bound

$$P(\|\frac{1}{m}Z^TZ - \Sigma^*\|_\infty > \epsilon) < 4p^2\exp\left\{-\frac{m\epsilon^2}{3200c^2_{\sigma^\star}}\right\} \tag{43}$$

for $0 < \epsilon < 40c_{\sigma^\star}$. □

Proof. (of Lemma 2)

The proof here proceeds exactly as in (Wainwright, 2009) and (Ravikumar et al., 2011), so we describe it relatively quickly. The goal is to show that when solve the restricted problem, the resulting $\tilde{\theta}$ (which is zero outside the support set S) is also optimal for the full ℓ_1 problem. Defining $\Delta = \tilde{\theta} - \theta^\star$, the the full ℓ_1 optimization problem can be written as

$$\nabla^2_\theta f(\theta^\star)\Delta + \nabla_\theta f(\theta^\star) - R(\Delta) + \lambda z = 0. \tag{44}$$

If we can show that $\|z\|_\infty < 1$, then $\tilde{\theta}$ is an optimal solution to the original ℓ_1 problem, so $\tilde{\theta} = \hat{\theta}$. Furthermore, the solution $\hat{\theta}$ cannot have support outside the support of $\theta^\star$.

We can write the above optimality condition in terms of S and $\bar{S}$, using $H = \nabla_\theta^2 f(\theta^\star)$ and $g = \nabla_\theta f(\theta^\star)$ for simplicity

$$\begin{bmatrix} H_{SS} & H_{S\bar{S}} \\ H_{\bar{S}S} & H_{\bar{S}\bar{S}} \end{bmatrix} \begin{bmatrix} \Delta_S \\ 0 \end{bmatrix} + \begin{bmatrix} g_S \\ g_{\bar{S}} \end{bmatrix} + \begin{bmatrix} R(\Delta)_S \\ R(\Delta)_{\bar{S}} \end{bmatrix} + \lambda \begin{bmatrix} z_S \\ z_{\bar{S}} \end{bmatrix} \tag{45}$$

Using the fact that

$$\Delta_S = H_{SS}^{-1}(R(\Delta)_S - g_S - \lambda z_S) \tag{46}$$

we can solve for$z_{\bar{S}}$ gives

$$z_{\bar{S}} = -\frac{1}{\lambda} H_{\bar{S}S}\Delta_S + \frac{1}{\lambda}(R(\Delta)_S - g_{\bar{S}}) = \frac{1}{\lambda} H_{\bar{S}S} H_{SS}^{-1}(g_S - R(\Delta)_S) + H_{\bar{S}S}H_{SS}^{-1} z_S + \frac{1}{\lambda}(R(\Delta)_S - g_S) \tag{47}$$

Thus

$$\|z_{\bar{S}}\|_\infty \leq \frac{2-\alpha}{\lambda}(\|g\|_\infty + \|R(\Delta)\|_\infty) + 1 - \alpha \leq \frac{2-\alpha}{\lambda}\frac{\alpha\lambda}{4} + 1 - \alpha < 1. \tag{48}$$

□

Proof. (of Lemma 3) Since $R(\Delta)$ is the residual of the first order Taylor expansion of the likelihood gradient, by the mean value theorem we have that the exists $t \in [0, 1]$ such that

$$R_\Lambda(\Delta_\Lambda, \Delta_\Theta) = d^2(\nabla_\Lambda f(\Lambda^\star + t\Delta_\Lambda, \Theta^\star + t\Delta_\Lambda); \Delta_\Lambda, \Delta_\Theta) \tag{49}$$

and similarly for $R_\Theta(\Delta_\Lambda, \Delta_\Theta)$. The first and second differentials of these gradient terms are given by (note that since these are the differentials of a matrix-valued function, we cannot simplify as many of the expressions as we did for the differential of the likelihood function)

$$\begin{aligned} d(\nabla_\Lambda f(\Lambda^\star, \Theta^\star); U, V) &= \Lambda^{-1}U\Lambda^{-1} + \Lambda^{-1}U\Lambda^{-1}\Theta^T S_{xx}\Theta\Lambda^{-1} + \Lambda^{-1}\Theta^T S_{xx}\Theta\Lambda^{-1}U\Lambda^{-1} - \\ &\quad \Lambda^{-1}V^T S_{xx}\Theta\Lambda^{-1} - \Lambda^{-1}\Theta^T S_{xx}V\Lambda^{-1} \\ d^2(\nabla_\Lambda f(\Lambda^\star, \Theta^\star); U, V) &= -2\Lambda^{-1}U\Lambda^{-1}U\Lambda^{-1} - 2\Lambda^{-1}U\Lambda^{-1}U\Lambda^{-1}\Theta^T S_{xx}\Theta\Lambda^{-1} - \\ &\quad 2\Lambda^{-1}U\Lambda^{-1}\Theta^T S_{xx}\Theta\Lambda^{-1}U\Lambda^{-1} - 2\Lambda^{-1}\Theta^T S_{xx}\Theta\Lambda^{-1}U\Lambda^{-1}U\Lambda^{-1} + \\ &\quad 2\Lambda^{-1}U\Lambda^{-1}V^T S_{xx}\Theta\Lambda^{-1} + 2\Lambda^{-1}U\Lambda^{-1}\Theta^T S_{xx}V\Lambda^{-1} + \\ &\quad 2\Lambda^{-1}V^T S_{xx}\Theta\Lambda^{-1}U\Lambda^{-1} + 2\Lambda^{-1}\Theta^T S_{xx}V\Lambda^{-1}U\Lambda^{-1} - \\ &\quad 2\Lambda^{-1}V^T S_{xx}V\Lambda^{-1} \\ d(\nabla_\Theta f(\Lambda^\star, \Theta^\star); U, V) &= -2S_{xx}\Theta\Lambda^{-1}U\Lambda^{-1} + 2S_{xx}V\Lambda^{-1} \\ d^2(\nabla_\Theta f(\Lambda^\star, \Theta^\star); U, V) &= 4S_{xx}\Theta\Lambda^{-1}U\Lambda^{-1}U\Lambda^{-1} - 4S_{xx}V\Lambda^{-1}U\Lambda^{-1} \end{aligned} \tag{50}$$

For example, $R_\Lambda(\Delta_\Lambda, \Delta_\Theta)$ is equal to the second expression with the $\Lambda^\star$ terms replaced by $\Lambda^\star + \Delta_\Lambda$, the $\Theta^\star$ terms replaced by $\Theta^\star + \Delta_\Theta$ and U and V replaced by Δ_Λ and Δ_Θ respectively. To bound $R(\Delta)$, we bound each of these terms individually.

Although the expression is rather lengthy, note that each term in the second differentials has a quadratic expression in Δ_Λ and Δ_θ, with at most four $(\Lambda^\star + t\Delta_\Lambda)^{-1}$ terms, two $\Theta^\star + t\Delta_\Theta$ terms and one S_{xx} term. Furthermore, we use the fact that

$$\|ABC\|_\infty = \|(C^T \otimes A)\operatorname{vec}(B)\|_\infty \leq |||C|||_1 |||A|||_\infty \|B\|_\infty \tag{51}$$

to place the vector infinity norm around the S_{xx} term in all cases, since $\|S_{xx}\|_\infty \leq c_X^2$. Thus, each term in the second differential is bounded by

$$c_X^2 |||(\Lambda^\star + t\Delta_\Lambda)^{-1}|||_\infty^4 |||\Theta^\star + t\Delta_\Theta|||_1^2 |||\Delta|||_1 \tag{52}$$

Now, first note that since Δ as a most d entries per column

$$|||\Delta|||_1 \leq d\|\Delta\|_\infty. \tag{53}$$

Now, note that

$$(\Lambda^\star + t\Delta_\Lambda)^{-1} = (I + t{\Lambda^\star}^{-1}\Delta_\Lambda)^{-1} \tag{54}$$

and

$$(I + t{\Lambda^\star}^{-1}\Delta_\Lambda)^{-1} = \sum_{i=1}^{\infty}(-1)^i(t{\Lambda^\star}^{-1}\Delta_\Lambda)^i \tag{55}$$

so that

$$\begin{aligned} \vert\vert\vert(\Lambda^\star + t\Delta_\Lambda)^{-1}\vert\vert\vert_\infty &\le \vert\vert\vert{\Lambda^\star}^{-1}\vert\vert\vert_\infty \sum_{i=1}^{\infty} \vert\vert\vert{\Lambda^\star}^{-1}\vert\vert\vert_\infty^i \vert\vert\vert\Delta_\Lambda\vert\vert\vert_\infty \|^i \\ &\le \frac{c_{\Sigma^\star}}{1 - c_{\Sigma^\star} d\|\Delta\|_\infty} \le \frac{3c_{\Sigma^\star}}{2}. \end{aligned} \tag{56}$$

Furthermore,

$$\vert\vert\vert\Theta^\star + t\Delta_\Theta\vert\vert\vert_1 \le \vert\vert\vert\Theta^\star\vert\vert\vert_1 + \vert\vert\vert\Delta_\Theta\vert\vert\vert_1 \le c_{\Theta^\star} + \frac{1}{2}d\|\Delta\|_\infty \le \frac{3c_{\Theta^\star}}{2}. \tag{57}$$

Combining these expressions results in the bound

$$\|R(\Delta)\|_\infty \le 206 c_{\Sigma^\star}^4 c_{\Theta^\star}^2 c_X^2 d^2 \|\Delta\|_\infty^2 \tag{58}$$

as required. □

Proof. (of Lemma 4) Let $(\Lambda^\star, \Theta^\star)$ be the true parameters with support S and $(\tilde{\Lambda}, \tilde{\Theta})$ be the solution to the optimization problem restricted to this support set. Our goal is to bound $\|\Delta\|_\infty$ where $\Delta = [\Delta_\Lambda \Delta_\Theta]$ with $\Delta_\Lambda = \tilde{\Lambda} - \Lambda^\star$ and $\Delta_\Theta = \tilde{\Theta} - \Theta^\star$.

Define

$$r := 2c_{H^\star}(\max\{\|\nabla_\Lambda f(\Lambda^\star, \Theta^\star)\|_\infty, \|\nabla_\Theta f(\Lambda^\star, \Theta^\star)\|_\infty\} + \lambda) \tag{59}$$

and note that by assumption we have

$$r \le 2c_{H^\star}(\min\left\{\frac{1}{3c_{\Sigma^\star}d}, \frac{1}{412 c_{\Sigma^\star}^4 c_{\Theta^\star}^2 c_X d^2}\right\}) \tag{60}$$

To bound $\|\Delta\|_\infty$ observe that we have $\Delta_C = 0$ and

$$\Delta_S = H_{SS}^{\star -1}(R_S(\Delta) + G_S - \lambda Z_S) \tag{61}$$

as shown in the proof for Lemma 2. Our approach will be the same as that of (Ravikumar et al., 2011), using Brouwer's fixed point theorem. To do so, note that we can view the RHS of the above equation as a continuous function of Δ and thus by Brouwer's fixed point theorem on a compact set, it suffices to show that if show that if $\|\Delta_S\|_\infty \le r$ then $\|H_{SS}^{*-1}(R_S - G_S - \lambda Z_S)\|_\infty \le r$ as this implies that there is a solution to this equation such that $\|\Delta_S\| \le r$ and by uniqueness (from Assumption 2) this solution must be $(\tilde{\Lambda}, \tilde{\Theta})$.

Taking infinity norm, we have

$$\|\Delta_S\|_\infty \le \|H_{SS}^{*-1}\|_\infty \|R(\Delta)\|_\infty + \|H_{SS}^{*-1}\|_\infty \|G_S - \lambda Z_S\|_\infty \tag{62}$$

For the first term, through application of the bound on $R(\Delta)$ and by assumption on $\|\Delta\|_\infty$

$$\|H_{SS}^{*-1}\|_\infty \|R(\Delta)\|_\infty \le 206\kappa_{H^*} c_{\Sigma^\star}^4 c_{\Theta^\star}^2 c_X d^2 \|\Delta\|_\infty^2 \le \frac{r}{2} \tag{63}$$

And for the second term

$$\|H_{SS}^{*-1}\|_\infty \|G_S - \lambda Z_S\|_\infty \le \kappa_{H^*}(\|G\|_\infty + \lambda) \le \frac{r}{2} \tag{64}$$

and thus the claim is proven. □

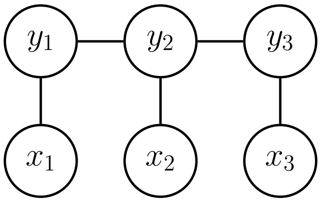

Figure 1. The chain CRF with 3 input variables and 3 output variables.

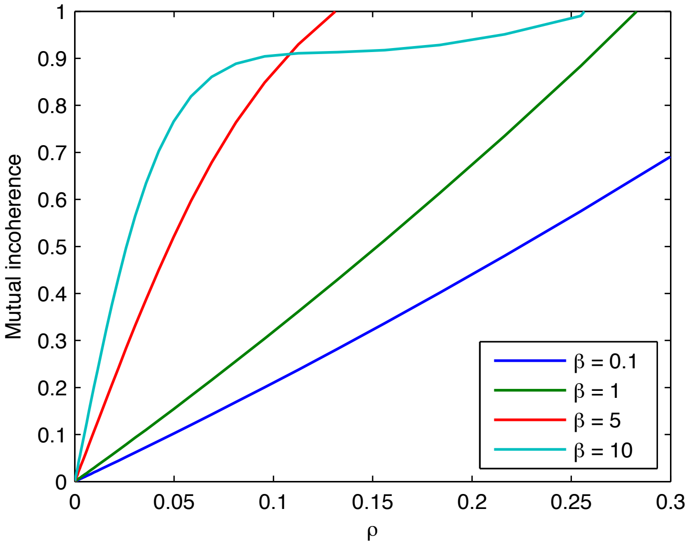

Figure 2. The mutual incoherence condition $\|\!|H_{\bar{S}S}(H_{SS})^{-1}\|\!|_\infty$ while varying ρ and β.

D. Mutual Incoherence for the Chain CRF

In this section we consider the mutual incoherence condition for the chain CRF, illustrated in Figure 1. For simplicity, we consider a class of models parameterized by two variables: ρ describing the conditional dependence between the output variables and β describing the relative influence of the input variables on the output variables. In particular, the class of models specified by

$$\Lambda^\star = \begin{bmatrix} 1 & \rho & 0 \\ \rho & 1 & \rho \\ 0 & \rho & 1 \end{bmatrix} \quad \Theta^\star = \begin{bmatrix} \rho\beta & 0 & 0 \\ 0 & \rho\beta & 0 \\ 0 & 0 & \rho\beta \end{bmatrix} \tag{65}$$

with positive ρ and β.

We are interested in characterizing the range over which the mutual incoherence condition

$$\|\!|H_{\bar{S}S}(H_{SS})^{-1}\|\!|_\infty < 1 \tag{66}$$

holds. Note that the Hessian (given in Appendix A) depends not only on these parameters, but also on the empirical covariance of the input features, S_{xx}; for the purpose of this illustration, we take S_{xx} to be the identity matrix, representing an ideal case in which the input features are perfectly uncorrelated. Under these conditions, we can see from Figure 2 that mutual incoherence indeed holds over a range of the parameters. However, as ρ increases and the output variables become more correlated, we approach the boundary at which this assumption is no longer valid.

References

Hsieh, C.-J., Sustik, M. A., Dhillon, I. S., and Ravikumar, P. Sparse inverse covariace matrix estimation using quadratic approximation. In *Neural Information Processing Systems*, 2011.

Magnus, X and Neudecker, Heinz. Matrix differential calculus. *New York*, 1988.

Ravikumar, Pradeep, Wainwright, Martin J, Raskutti, Garvesh, and Yu, Bin. High-dimensional covariance estimation by minimizing 1-penalized log-determinant divergence. *Electronic Journal of Statistics*, 5:935–980, 2011.

Wainwright, M.J. Sharp thresholds for high-dimensional and noisy sparsity recovery using ℓ_1-constrained quadratic programming (Lasso). *IEEE Transactions on Information Theory*, 55(5):2183–2202, 2009.