Extending the Modelling Capacity of Gaussian Conditional Random Fields while Learning Faster

Jesse Glass
Temple University
Philadelphia, USA
tud25892@temple.edu

Mohamed Ghalwash
Temple University
Philadelphia, USA
tuc30491@temple.edu

Milan Vukicevic
University of Belgrade
Belgrade, Serbia
vukicevicm@fon.bg.ac.rs

Zoran Obradovic
Temple University
Philadelphia, USA
zobrad@gmail.com

Abstract

Gaussian Conditional Random Fields (GCRF) are a type of structured regression model that incorporates multiple predictors and multiple graphs. This is achieved by defining quadratic term feature functions in Gaussian canonical form which makes the conditional log-likelihood function convex and hence allows finding the optimal parameters by learning from data. In this work, the parameter space for the GCRF model is extended to facilitate joint modelling of positive and negative influences. This is achieved by restricting the model to a single graph and formulating linear bounds on convexity with respect to the models parameters. In addition, our formulation for the model using one network allows calculating gradients much faster than alternative implementations. Lastly, we extend the model one step farther and incorporate a bias term into our link weight. This bias is solved as part of the convex optimization. Benefits of the proposed model in terms of improved accuracy and speed are characterized on several synthetic graphs with 2 million links as well as on a hospital admissions prediction task represented as a human disease-symptom similarity network corresponding to more than 35 million hospitalization records in California over 9 years.

Modelling complex phenomena through instances that are highly structured and interdependent is a challenging task which traditional predictive modelling techniques cannot address efficiently. Many high impact applications have such properties and they are usually modelled by applying graphical structure learning theory. Still, state-of-the-art probabilistic graphical models often do not define convex parameter search space and thus, cannot guarantee global optimum and efficient search. Structured models for regression have been researched less than classification problems. But, several research teams within the past 5 years have independently proposed methods for prediction using Gaussian Conditional Random Fields.

It has been shown that if the relationship among the output is represented as a quadratic form, then the traditional Continuous Conditional Random Field model has the form of multivariate Gaussian distribution (hence the name Gaussian Conditional Random Fields - GCRF). This results in a convex parameter search space that guarantees global optimum and efficient learning and inference.

There exist two major approaches to GCRF algorithms: network link weight optimization algorithms and hidden network learning algorithms. Network link weight optimization algorithms incorporate link weights as a feature. For example, one could have a network weight indicating the number of times someone posts on their friend's Facebook page. These network weights are used to build a lens that shifts predictions in order to improve accuracy. The algorithm actually learns a new space in which it is able to make more accurate predictions. Optimizing the input predictions in the new space is similar to using Generalized Least Squares, but here we are also learning the space. This can be seen in (Radosavljevic, Vucetic, and Obradovic 2010).

The other group of algorithms, hidden network learning, focus on directly updating the precision matrix. Instead of incorporating link weights it learns pairwise connections that affect the final prediction. This group of algorithms has shown success on data where networks are known to exist yet no network data is available. This is by its nature more restrictive on input data and the resulting learned relationships are less interpretable than relationships optimized in link weight feature GCRF. An example of this algorithm class can be found in (Wytock and Kolter 2012). And a more general framework for hidden network learning is documented in (Lee and Hastie 2013).

These two different groups of algorithms are for different purposes. For example, there is a growing area of biomedical research where investigators survey biomedical literature to construct disease similarity networks (Zhou et al. 2014). Incorporating such a disease similarity network, which cannot be represented in traditional formats for many machine learning methods, is a valuable step forward. Hence, the focus of this paper will be on link weight feature GCRF.

The network operator in GCRF acts as a smoothing function. But sometimes over-smoothing can occur. One can include de-smoothing weights between targets so as to push their values away from one another with the extensions to the parameter space shown in this paper.

In this research we propose a new algorithm that introduces a mathematical formulation that extends the GCRF parameter space to include negative values Unimodal GCRF (UmGCRF). The requirement is that we maintain positive

definiteness of the precision matrix. We change the algorithm from that proposed in (Radosavljevic, Vucetic, and Obradovic 2010) by restricting the model to one network, thus the name. In the presented formulation, we can write our boundary conditions for positive definiteness as a set of linear equations of our parameters. This is the ideal for constrained gradient descent and allows negative parameter values into the convex optimization space for this problem. Our mathematical formulation also yields a dramatic speed up. Last, we introduce a network link weight bias term as part of the convex optimization.

Related Work

This research expands on the model of (Radosavljevic, Vucetic, and Obradovic 2010) by allowing a broader range of both linear combinations of unstructured predictors and network structures. The previous method could only take a weighted average of both unstructured predictors and link weights. It also required that all link weights be positive. As a result of restricting the number of graphs that we use, we can establish exact boundaries as linear function of the parameters.

The hidden network learning algorithm that we compare with is Sparse GCRF (SpGCRF) developed in (Wytock and Kolter 2012). In that paper, they ensure that the precision matrix, Q, is positive definite by a common technique of simply defining $log|Q|$ to be infinite if $Q \succ 0$. This seems to be a limited approach.

A benefit to using input link weights – beyond the opportunity to incorporate otherwise overlooked information – is that the the method is faster and has greater scalability than network learning algorithms. For a comparison, we look at four implementations. SpGCRF was chosen to represent network learning algorithms because they made code available for testing. We also compare the time for the original proposed method, GCRF, and a fast learning approximation method, FF-GCRF (Ristovski et al. 2013). The following speed tests were done in Matlab with a single feature per target variable.

Target Size	UmGCRF	FF-GCRF	GCRF	SpGCRF
1,000	4.7 secs	3.3 secs	41 secs	13.5 mins
5,000	43 secs	34 secs	9.2 mins	28.1 hours
10,000	3.9 mins	2.9 mins	1.76 hours	9.7 days
20,000	30.2 mins	22.4 mins	17 hours	N/A
40,000	5 hours	3 hours	7.5 days	N/A
100,000	21 hours	16 hours	N/A	N/A

Table 1: speed of different algorithms

We can see that GCRF far exceeds the speed and scalability of SpGCRF. The approach presented in this paper is nearly as fast as FF-GCRF, which is an approximation technique developed by (Ristovski et al. 2013). The approach used to method from (Adams, Baek, and Davis 2010) to approximate inference and differentiation. But since this method provides approximations of optimal GCRF parameters and approximate inference, it has also produced lower accuracy and is therefore out of the scope of this paper.

In terms of the current research using GCRF or FF-GCRF the results from research on Climate (Radosavljevic, Vucetic, and Obradovic 2010; 2014; Djuric et al. 2015), Energy forecasting (Guo 2013), Healthcare (Gligorijevic, Stojanovic, and Obradovic 2015; Polychronopoulou and Obradovic 2014) are based on using a single network. Since this is the only restriction on our new implementation, it is safe to assert that our restriction to one network is not a substantial loss.

For the application presented at the end of this paper, we incorporate a disease similarity network developed in (Zhou et al. 2014). The authors used biomedical literature database to construct a symptom-based human disease network. This is a feature space that cannot be represented in traditional formats for machine learning. And, this type of information is a growing area of biomedical research.

Methods

In regression on graphs, a vector of attributes $\mathbf{x}$ and a real-valued response variable $\mathbf{y}$ are observed at previous time steps at nodes of a graph while the objective is to predict future value of $\mathbf{y}$ at all nodes given features $\mathbf{x}$. The GCRF is a discriminative model for regression on an attributed evolving graph that models the conditional distribution $P(\mathbf{y}|\mathbf{x})$ over N nodes for outputs $\mathbf{y}$ given the corresponding inputs $\mathbf{x}$:

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x},\alpha,\beta)} \exp\Big(\sum_{i=1}^{N} A(\alpha, y_i, \mathbf{x}) + \sum_{j\sim i} I(\beta, y_i, y_j)\Big)$$

where α and β are parameters of the association A and the interaction I potentials, respectively, and the normalization term $Z(\mathbf{x},\alpha,\beta)$ is an integral over $\mathbf{y}$ of the term in the exponent. The association potential function is defined as (Radosavljevic, Vucetic, and Obradovic 2010):

$$A(\alpha, y_i, \mathbf{x}) = -\sum_{i=1}^{N}\sum_{k=1}^{K} \alpha_k (y_i - R_k(\mathbf{x}))^2$$

where $R_k(\mathbf{x})$ represents any function that maps $\mathbf{x} \rightarrow y_i$ for each node in the graph. We refer to this function as unstructured predictor (any regression model) that gives independent predictions. The influence of each unstructured predictor R_k on the final predicted value is modelled by GCRF by optimizing parameters α_k, where K is the number of unstructured predictors. The interaction potential function is defined as:

$$I(\beta, y_i, y_j) = -\sum_{l=1}^{L}\sum_{i\sim j} \beta_l S_{ij}^l (y_i - y_j)^2$$

The similarity between two nodes i and j is defined as S_{ij}^l. The GCRF model ensures that the prediction of two similar nodes are similar. This influence of the similarity (and hence of the structure of the graph) is modelled through the interaction potential and weighted by the parameter β_l, where L is the number of similarity functions (multi-modal graph).

The conditional probability model can be rewritten as:

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x},\alpha,\beta)} \exp\Big(-\sum_{i=1}^{N}\sum_{k=1}^{K} \alpha_k (y_i - R_k(\mathbf{x}))^2 - \sum_{l=1}^{L}\sum_{i\sim j} \beta_l S_{ij}^l (y_i - y_j)^2\Big)$$

GCRF canonical form. Modelling association and interaction potentials as quadratic functions of $\mathbf{y}$ enables GCRF to represent CRFs as multivariate Gaussian distribution (Radosavljevic, Vucetic, and Obradovic 2010):

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{(2\pi)^{\frac{N}{2}}|\Sigma|^{\frac{1}{2}}} \exp\Big(-\frac{1}{2}(\mathbf{y}-\mu)^T\mathbf{Q}(\mathbf{y}-\mu)\Big)$$

where $\Sigma^{-1}(:= 2\mathbf{Q})$ is the inverse covariance matrix:

$$\mathbf{Q} = \begin{cases} \sum_k \alpha_k + \sum_h \sum_l \beta_l S_{ih}^l & \text{if } i = j \\ -\sum_l \beta_l S_{ij}^l & \text{if } i \neq j \end{cases}$$

Inference. The inference task $argmax_{\mathbf{y}} P(\mathbf{y}|\mathbf{x})$ is straightforward. Since GCRF is represented as multivariate Gaussian distribution, the maximum posterior estimate of $\mathbf{y}$ is obtained by computing the expected value $\mu = \mathbf{Q}^{-1}\mathbf{b}$, where $b_i = 2\sum_k \alpha_k R_k(\mathbf{x})$.

Learning. The learning objective is to optimize the parameters α and β by maximizing the conditional log–likelihood

$$\underset{\alpha,\beta}{argmax} \sum_{\mathbf{y}} \log P(\mathbf{y}|\mathbf{x})$$

To ensure the feasibility of the GCRF model, the $\mathbf{Q}$ matrix must be positive definite. All previous implementations set constraints on the parameters so that $\alpha > 0$ and $\beta > 0$. But, this unnecessarily limits the search space and makes GCRF unable to incorporate negative links, nor to identify negative influence of unstructured predictors. In the next section, we will expand the search space of the parameters to relax these assumptions.

Contribution. The first observation is that the $\mathbf{Q}$ can be written more concisely as $Q = \sum_k \alpha_k I + \sum_j \beta_j L_j$. Here L_j is the Laplacian of the matrix S_j. For our model we focus on the case where there is only one similarity network, so $Q = \sum_k \alpha_k I + \beta L$. Next, we examine the effect of diagonalizing Q. We know $L = UDU^T$ where $UU^T = I$ and D is a diagonal matrix because L is a symmetric real valued matrix.

$$\begin{aligned} Q &= \textstyle\sum_k \alpha_k I + \beta L = \sum_k \alpha_k I + \beta U D U^T \\ &= U(\,(\textstyle\sum_k \alpha_k)\cdot U^T I U + \beta D) U^T \\ &= U(\textstyle\sum_k \alpha_k I + \beta D) U^T \end{aligned}$$

Then, $Q = U\Lambda U^T$ where Λ is diagonal matrix, with diagonal elements:

$$\lambda_i = \sum_k \alpha_k + \beta d_i \quad \forall i \tag{1}$$

We will provide exact bounds for convexity for the unimodal case. We establish that our parameter search space is convex by showing that covariance matrix is Positive Definite subject to our boundary conditions.

Lemma [uni-modal]:

$$\mathbf{Q} \succ 0 \Leftrightarrow \begin{cases} \sum_k \alpha_k + \beta d_0 > 0 \\ \sum_k \alpha_k + \beta d_{n-1} > 0 \end{cases}$$

Start with a theorem established in (Ayres 1967): Given a real symmetric matrix, Q, $\exists\ U$ such that $Q = U\Lambda U^T$ and $Q \succ 0 \Leftrightarrow \lambda_i > 0\ \forall i$ where λ_i are the diagonal entries in Λ. Next, substitute λ_i with a function in terms of our parameters: $\lambda_i = \sum_k \alpha_k + \beta d_i$ (1).
Since D is a diagonalized matrix we know that d_i are in ascending order, d_0 being lowest and d_{n-1} being highest. As a result,

$$\beta d_{n-1} \geq \ldots \geq \beta d_0 \text{ if } \beta \geq 0$$
$$\beta d_{n-1} \leq \ldots \leq \beta d_0 \text{ if } \beta \leq 0$$

Since $\sum_k \alpha_k$ effects each diagonal equally, $\forall\beta$, each diagonal entry in Λ is in between λ_{n-1} and λ_0. Thus, only the outermost constraints are required to ensure positive definiteness. ■

With linear boundary conditions, the optimization can be done with an interior point bounded gradient descent. Below are graphical representations of the new parameter search space. Previously, all searches were restricted to the first quadrant. In our case, if $d_0 = 0$ then we search the entire first quadrant and additional space.

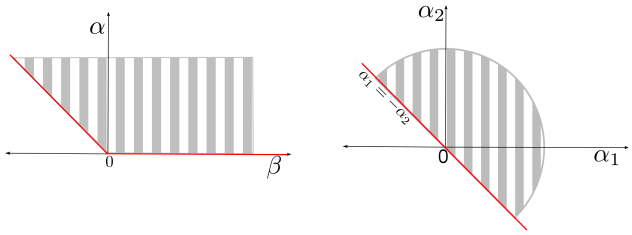

Figure 1: Parameter Space for β & $\sum_i \alpha_i$ (left) and α (right).

In order to demonstrate the additional modelling capacity of the new parameter search space, we walk through a simple example case. In this case there are only two targets. The right panel in Figure 2 shows the smoothing behavior of traditional GCRF pulling updated predictions towards each other, but away from their true value. The left panel shows the behavior possible as a result of negative links or negative betas. We can now push values away from each other, and in this toy case toward their true values.

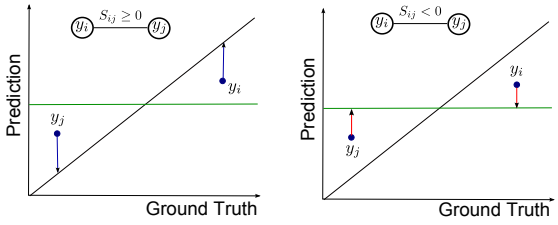

Figure 2: Similarity: $\beta S_{ij} \leq 0$ (left) & $\beta S_{ij} \geq 0$ (right)

Speed. Previous approaches required matrix inversion in order to calculate the first order derivatives. This slowed down

gradient descent at each iteration by $O(n^3)$. Additionally, the previous methods inferred μ at every iteration as input to calculate the first order derivatives, a cost of $O(n^2)$.

$\frac{\partial l}{\partial \alpha_i} = \frac{-1}{2}(y^T y + 2R_i^T(\mu - y) + \mu^T\mu) + \frac{1}{2}Tr(Q^{-1})$

$\frac{\partial l}{\partial \beta} = \frac{-1}{2}(y^T L y + \mu^T L \mu) + \frac{1}{2}Tr(Q^{-1}L)$

Only the trace of the inverted matrix is needed, so solving for the eigenvalues directly makes the same calculation $O(n)$. We side-stepped the need to infer μ for our first order derivatives. So after we eigendecompose the Laplacian of the Similarity matrix before gradient descent, each iteration only takes $O(n)$ operations. In order to demonstrate this, we replace operations that can occur outside gradient descent with x symbols and use $\times$ to indicate element-wise multiplication. Also note the step, $C = U^T R$ that can occur outside the gradient update procedure.

$\frac{\partial l}{\partial \alpha} = \frac{-1}{2}(x_1 + 2(C_i \times \lambda^{-1})C\alpha + \alpha^T C^T \Lambda^{-2} C\alpha) - 1^T\lambda^{-1})$

$\frac{\partial l}{\partial \beta} = \frac{-1}{2}(x_2 - \alpha^T C^T((d \times \lambda^{-2})1^T \times C)\alpha) + \frac{1}{2}d^T\lambda^{-1}$

Additional Rank One Matrix. In order to expand the power and scope of our model we will include an additional rank one similarity network that will serve as a bias term for network weights. This will help us when we want to incorporate networks with non-negative weights whether it's a result of other people's research or using similarities such as Gaussian Kernels or Cosine Similarity. By introducing an intercept Matrix $J = \vec{1}\vec{1}^T$, we can shift origin for similarity weights so that they are centered around any chosen point. This could guide researchers' understanding of network weights in the future. We note here that the optimization for this shift is convex. This new method maintains the speed established above and includes more modelling capacity. Here, we note that the Laplacian of $\frac{1}{n}J$ is $I - \frac{1}{n}J$. Consequently, $Q = \sum_k \alpha_k I + \beta L + \varsigma I - \varsigma\frac{1}{n}J$.

Laplacian matrices have been studied in depth. In (Merris 1998), it was proven that all Laplacian matrices have an eigenvector of $v = \vec{1}/|\vec{1}|$ and an associated $\lambda = 0$. In (Ding and Zhou 2007), it was shown that the perturbation of a matrix by a rank one matrix that has a basis in the original matrix only contributes to the the eigenvalue associated with that basis. Since we know that $v = \vec{1}/|\vec{1}| \in U$ this implies $U^T\frac{1}{n}JU = \frac{1}{n}(U^T\vec{1})(U^T\vec{1})^T = D_j$, a matrix of zeros with a one on the diagonal entry associated with $col(U) = v$. With this established, we can diagonalize Q.

$$\begin{aligned} Q &= \textstyle\sum_k \alpha_k I + \beta L + \varsigma I - \varsigma\frac{1}{n}J \\ &= \textstyle\sum_k \alpha_k I + \beta U D U^T + \varsigma I - \varsigma\frac{1}{n}J \\ &= U((\varsigma + \textstyle\sum_k \alpha_k)\cdot U^T I U + \beta D - \frac{\varsigma}{n}U^T J U)U^T \\ &= U(\textstyle\sum_k \alpha_k I + \varsigma I + \beta D - \varsigma D_J)U^T. \end{aligned}$$

Since we now have this bias term, ς/β, we can shift all weights into the non-negative space and still map the original values in addition to a broader range of values.

$$I(\beta, y_i, y_j, \varsigma) = -\sum_{l=1}^{L}\sum_{i\sim j}(\beta_l S_{ij}^l + \varsigma)(y_i - y_j)^2$$

Having strictly non-negative weights makes notation for the following notation simpler but it is not necessary that weights be non-negative. For Laplacian matrices with non-negative similarity weights, we know that $\vec{1}/|\vec{1}|$ is associated with the lowest eigenvalue, λ_0. Then, $Q = U\Lambda U^t$ where Λ is diagonal matrix, with diagonal elements:

$$\begin{cases} \lambda_i = \sum_k \alpha_k + \varsigma + \beta d_i \text{ if } i \neq 0 \\ \lambda_0 = \sum_k \alpha_k + \beta d_0 \end{cases} \quad (2)$$

Theorem [uni-modal plus rank one]:

$$\mathbf{Q} \succ 0 \Leftrightarrow \begin{cases} \sum_k \alpha_k + \beta d_0 > 0 \\ \sum_k \alpha_k + \varsigma + \beta d_1 > 0 \\ \sum_k \alpha_k + \varsigma + \beta d_{n-1} > 0 \end{cases}$$

We use the theorem established in (Ayres 1967) to state $Q \succeq 0 \Leftrightarrow \lambda_i > 0 \forall i$ where λ_i. Substitute lambda for a function with respect to the models parameters (eq.2). Since D is a diagonalized matrix we know that d_i are in ascending order, d_0 being lowest and d_{n-1} being highest. As a result,

$$\beta d_{n-1} \geq ... \geq \beta d_0 \text{if } \beta \geq 0$$
$$\beta d_{n-1} \leq ... \leq \beta d_0 \text{if } \beta \leq 0$$

Thus λ_{n-1} through λ_1 are greater than zero so long as:

$$\textstyle\sum_k \alpha_k + \varsigma + \beta d_1 > 0$$
$$\textstyle\sum_k \alpha_k + \varsigma + \beta d_{n-1} > 0$$

This leaves us with a final constraint $\sum_k \alpha_k + \beta d_0 > 0$ and we have established bounds on positive definiteness for Q. ■

These new bounds remain linear with respect to our parameters and the first order derivative remain unchanged for α and β. The derivative with respect to ς is also linear cost so we did not slow down our gradient calculations. When experimenting with UmGCRF, we found that a regularization component was helpful to reduce testing error. So in the final version, we included a basic weight decay for ς in the likelihood function.

Experimental evaluation

We started the evaluation with myriad synthetic datasets. First, examine a case where we expect UmGCRF and GCRF to perform similarly ($\alpha, \beta > 0$). Next, we look at cases where $\alpha > 0$ & $\beta < 0$ and $\alpha_1 > 0$ & $\alpha_2 < 0$. The last synthetic experiment is a series of trials used to compare GCRF and UmGCRF performance depending on the number of negative link weights in the network. Last, we performed a comparison of methods on a healthcare task aimed at predicting the monthly number of admissions by disease for hospitals in California.

Evaluation on Synthetic Graphs

In the following three experiments, we generated a vector of length 2000 which is then used as an unstructured prediction. We generated a positive valued matrix which represents a uni-model graph with 2000 nodes and 2 million edges. Then, choosing appropriate values for α and β we generated our target vector using the GCRF model. Models were compared in terms of conditional log-likelihood (LL) and R^2.

The synthetic data for this experiment was generated with parameters $\hat{\alpha} = 0.792$ and $\hat{\beta} = 0.61$. The results of regression on such a graph are shown in Table 2. In this experiment UmGCRF and GCRF models were nearly identical and almost perfect accuracy.

Table 2: Positive influence. LL = conditional log-likelihood.

Model	LL	α	β	R^2
GCRF	-97.346	0.791	0.611	0.999
UmGCRF	-97.350	0.793	0.609	0.999

The next synthetic dataset was generated with parameters $\hat{\alpha} = 1$ and $\hat{\beta} = \frac{-1}{2}$. The results of regression on that graph by UmGCRF vs. GCRF shown in Table 3 provide evidence that UmGCRF significantly outperforms GCRF.

Table 3: Negative influence of links.

Model	LL	α	β	R^2
GCRF	-5.83e+05	0.820	0.573	0.31
UmGCRF	-1.11e+03	0.894	-0.447	**0.56**

Here, the synthetic data was generated using two unstructured predictors with the optimal parameters $\hat{\alpha_1} = \frac{\sqrt{3}}{2}$ and $\hat{\alpha_2} = \frac{-1}{2}$. GCRF could not find the optimal negative parameter α_2 and instead it found a positive value (which is sub-optimal) that affects its accuracy as indicated by a very low R^2.

Table 4: Influence of unstructured predictor.

Model	LL	α_1	α_2	R^2
GCRF	-54.21e+03	0.721	0.693	0.051
UmGCRF	-01.37e+03	0.866	-0.500	**0.78**

That scenario is illustrated at Figure 3. In the left panel, the expanded search for α_1 and α_2 is projected onto the unit (half) circle. In the right panel, we plot the angle between the two parameters (x-axis) and the corresponding normalized negative log-likelihood NLL (y-axis). The white region at the right panel corresponds to the first quadrant at the left panel where both parameters α_1 and α_2 are positive and this is the space where GCRF looks for the optimal parameter. Clearly GCRF can not find the optimal parameter (red vertical line at the yellow region in the right panel) because they are out of its parameter space, while UmGCRF searches for the optimal parameters in all 3 regions (green, yellow, and white) which can be found.

Positive/Negative Influence Links The reason that UmGCRF significantly outperformed GCRF in Table 3 is that the links (similarity) have negative influence on the predicted value which cannot be captured by GCRF, but were captured by UmGCRF model. In real-life applications, it is not necessary the case that all links have either positive (Table 2) or negative (Table 3) influence on the prediction.

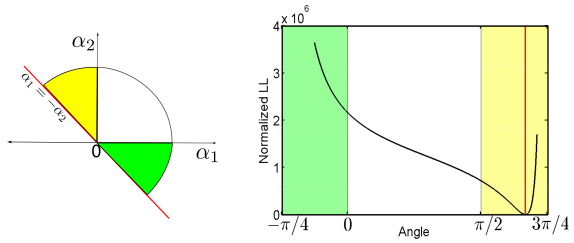

Figure 3: Optimal parameters outside the original search space

In many cases (as depicted in many real datasets including the one described in the next section) some of the links might have positive influence and some other links might have negative influence. To evaluate benefit of UmGCRF in this scenario we generated 7 synthetic graphs with $0\%, 16\%, 35\%, 50\%, 65\%, 84\%$, and 100% of positive inks, respectively. Here, the graph that corresponds to 0% is basically the experiment shown in Table 3 while 100% corresponds to the experiments summarized in Table 2. The regression results by UmGCRF on these 7 datasets for these experiments are shown in Figure 5.

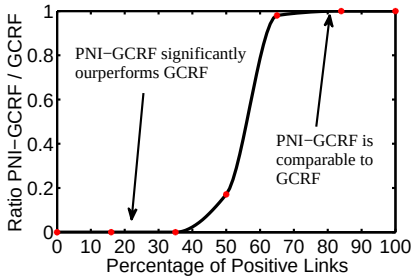

Figure 4: Ratio of MSE of UmGCRF to GCRF for different datasets with different percentages of positive links.

The x-axis in Figure 4 represents the percentage of positive links the graph while the y-axis represents the ratio of Mean Square Error (MSE) of UmGCRF to GCRF. So, if UmGCRF significantly outperforms GCRF then the ratio tends to zero, while the ratio tends to one when both models have equally performance.

Prediction of Hospital Admissions

We evaluated UmGCRF on the problem of predicting monthly hospital admissions for 189 classes of diseases in California from HCUP data (HCUP 2011). The target is to predict the number of admission for each disease for the next month.

For each of 35,844,800 inpatient discharge records collected over 9 years in the California HCUP database there are up to 253 diagnosis codes in CSS coding schema. In this study, we constructed monthly disease graphs such that each node represents one disease. But 22 diseases had incomplete information over the time period analysed, resulting in 231

nodes. In this experiment we use the disease-symptom similarity network built in (Zhou et al. 2014). Since this network was build using MeSH terminology, we built a translation table by hand for CSS codes to the specific MeSH terminology used in (Zhou et al. 2014). That table is publicly available at http://astro.temple.edu/∼tud25892. The matching is not one-to-one. Sometimes we would map several MeSH terms to a single CSS code. In these cases, we would take the average of similarities. There were also 52 CSS codes with no MeSH term in the network used. This reduced the number diseases in our analysis to 189.

Assume that $x_{t,i}$ is the number of patients diagnosed with disease i during the month t. For each node i we computed the rate of admission change $y_{t,i} = (x_{t,i} - x_{t-1,i})/x_{t-1,i}$. We used 108 monthly graphs (representing years 2003-2011). After converting this to a rate of change, we have 107 time points. We train on the first 80 months and test on the remaining 27. We slide a window of length $w = 12$. We trained two unstructured predictors: Linear regression (LR) and a Neural Network (NN). They were used as input for both GCRF and UmGCRF. NN had 26 hidden nodes.

The algorithm was tested 100 times because the NN is non-convex and yields different results each time. That has an effect of altering the output of GCRF and UmGCRF. Averages are taken to report results.

Algorithm	Testing RMSE
LR	0.1714
NN	0.1661
GCRF	0.1559
UmGCRF	**0.1558**

Although the improvement in accuracy is marginal, UmGCRF outperforms GCRF, achieving a lower testing error for greater than 80% of trials in an experiment of a thousand trials. This increase in accuracy is due to an improved network model. The network input was disease similarity on a zero to one scale. Upon inspection of parameters, we see that our learned intercept term tended to be around -.05. That shifts the similarity scale from $[0, 1]$ to $[-0.05, 0.95]$.

Of the 100 trials used for the table above, UmGCRF outperformed GCRF 89 times on training data and 71 times on testing Data. UmGCRF also outperformed NN on training Data and on testing Data 95 times as we can see in Figure 5. UmGCRF had 17% and 12% improvement in test accuracy over input baselines.

This small improvement in accuracy over GCRF represents improving predictions by 57 patients per month. Closer inspection of the results showed that UmGCRF had better performance on predicting 180 of 189 diseases. In Table we show the error reduction and true patient count for three diseases on which we identified the largest difference in accuracy measured in patients.

Conclusion

GCRF is a powerful tool that captures the graph structure in order to improve regression accuracy. However, GCRF is restricted to positive weights in order to preserve the positive semi-definiteness of the precision matrix. This imposes constraints in the parameter space to ensure the feasibility of the model. In this paper, we expanded the parameter search space to allow for negative links and negative influence of the unstructured predictors while maintaining the positive semi-definiteness of the precision matrix and improve computational efficiency. Our results provide evidence that the new model outperforms the original GCRF and unstructured predictors on both synthetic and real world data.

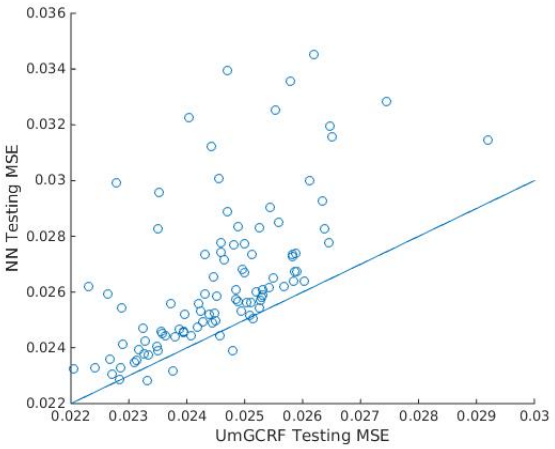

Figure 5: Scatter Plot of MSE for NN and UmGCRF, line represents equal

Disease	Accuracy Improvement Measured in # of Patients per Month
Spinal Cord Diseases	11
Heart Failure	8
Cerebrovascular Disorders	6

Table 5: Top 3 performance differences between UmGCRF and GCRF.

Acknowledgements

This research was supported in part by DARPA grant FA9550-12-1-0406 negotiated by AFOSR, NSF BIGDATA grant 14476570, ONR grant N00014-15-1-2729 and SNSF Joint Research project (SCOPES), ID: IZ73Z0_152415. Healthcare Cost and Utilization Project (HCUP), Agency for Healthcare Research and Quality, provided data used in this study.

References

Adams, A.; Baek, J.; and Davis, M. A. 2010. Fast high-dimensional filtering using the permutohedral lattice. In *EUROGRAPHICS*.

Ayres, F. 1967. Theory and problems of matrices. In *Theory and Problems of Matrices*.

Ding, J., and Zhou, A. 2007. Eigenvalues of rank-one updated matrices with some applications. In *Applied Mathematics Letters*.

Djuric, N.; Radosavljevic, V.; Obradovic, Z.; and Vucetic, S. 2015. Gaussian conditional random fields for aggregation of operational aerosol retrievals. *IEEE Geoscience and Remote Sensing Letters*.

Gligorijevic, D.; Stojanovic, J.; and Obradovic, Z. 2015. Improving confidence while predicting trends in temporal disease networks. In *4th Workshop on Data Mining for Medicine and Healthcare, SIAM International Conference on Data Mining (SDM)*.

Guo, H. 2013. Modeling short-term energy load with continuous conditional random fields. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD)*, 433–448.

HCUP. 2011. HCUP State Inpatient Databases (SID). Healthcare Cost and Utilization Project (HCUP). 2005-2009. Agency for Healthcare Research and Quality, Rockville, MD. http://www.hcup-us.ahrq.gov//sidoverview.jsp.

Lee, J., and Hastie, T. 2013. Learning the structure of mixed graphical models. In *Journal of Machine Learning Research*.

Merris, R. 1998. Laplacian graph eigenvectors. In *Linear Algebra and its Applications*.

Polychronopoulou, A., and Obradovic, Z. 2014. Hospital pricing estimation by gaussian conditional random fields based regression on graphs. In *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 564–567.

Radosavljevic, V.; Vucetic, S.; and Obradovic, Z. 2010. Continuous conditional random fields for regression in remote sensing. In *Proccedings of European Conference on Artificial Intelligence (ECAI)*, 809–814.

Radosavljevic, V.; Vucetic, S.; and Obradovic, Z. 2014. Neural gaussian conditional random fields. In *Proceeding of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD)*.

Ristovski, K.; Radosavljevic, V.; Vucetic, S.; and Obradovic, Z. 2013. Continuous conditional random fields for efficient regression in large fully connected graphs. In *AAAI*.

Wytock, M., and Kolter, Z. 2012. Sparse gaussian conditional random fields. In *NIPS workshop on log-linear models*.

Zhou, X. Z.; Menche, J.; Barabasi, A.-L.; and Sharma, A. 2014. Human symptoms-disease network. In *Nature Communications*.