Hierarchical Latent Space Models for Multiplex Social Network Analysis

Alex Martin Loewi
aloewi@cmu.edu

Francisco Ralston Fonseca
fralston@andrew.cmu.edu

Abstract

An extension of the Latent Space Model is developed, for parsimoniously fitting multigraph data. The model finds embeddings for all of the graph layers in a way that makes them visually comparable, a difficult to achieve but important property for social network analysis and many other fields including medical diagnostics, and deep learning. The model is fit using a group LASSO penalty that both increases interpretability, and also allows for graph-wise dimensionality reduction of this complicated form of data. Results are demonstrated on a novel data set of eighty individuals from the staff of a charter school.

1 Introduction

1.1 The Difficulty and Ubiquity of Multigraphs

Multigraphs, also called multiview networks, can be thought of as either a single graph with multiple edge types, or as multiple networks over the same set of nodes. Data of this kind is not uncommon, at either large or small scales. Twitter has Following relationships, as well as Retweets, and Likes, each of which can be thought of as simply one layer within the full relationship between two nodes. In small scales, surveys for networks very rarely ask about a single type of relationship, and many of the most famous network data sets have multiple relationships (such as Trust, Friendship, and Respect, in Samson's monk data). Despite the existence of data of this form for decades, its inherent difficulty has led to only a small number of models for it, all of which have important shortcomings for the practicing social network analyst.

1.2 The Practical Difficulties of Multigraphs

In particular, practitioners have two immediate problems when dealing with multigraphs, both of which are consequences of the inherent complexity of the data.

1.2.1 Visualization

The first problem is that multigraphs are too complex to easily visualize, and visualization has always been a cornerstone of network analysis. In particular, the difficulty comes in comparing the many layers of data (i.e. the graphs formed by the different edge types). To be expressive, a graph layout needs to be flexible in how it places nodes – however in order to be comparable, two layouts need to place nodes in relatively similar places. A useful multigraph model would be able to tune explicitly between these two extremes of flexibility, and comparability.

1.2.2 Dimensionality

The second problem is that multigraphs may be unnecessarily complex. Empirically, there are often high degrees of correlation between the layers in a multigraph, and the smaller the number of graphs,

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

the easier analysis becomes. These two observations combine to suggest another useful property of a new model – the ability to remove redundant layers.

1.3 Previous Work

Our model combines several methods that have proven to be exceptionally valuable – Latent Space Models, [10], the LASSO estimator, [2], and in particular the group LASSO [**?**]. The substantial amount of work that has been done on them all provides many different possible approaches to designing, and fitting, our own model. Efficient optimization procedures, often employing coordinate or block-coordinate optimization, exist for fitting the LASSO to linear models, [2], to generalized linear models [3], with the element-wise, group, and combined, sparse group LASSO, [5, 7, 8], and more.

When it comes to LSMs for multigraphs, this year already, two different papers have been published that take this general appraoch – however, our approach has important advantages over both of them. One paper takes the extreme of modeling the multigraph as a single object, which only makes visualization more difficult [**?**]. The other treats each layer as independent, and allows correlations between them. While a highly intuitive idea, this does not solve the comparability problem, or that of dimensionality reduction [**?**]. Furthermore, allowing correlations between layers means that an edge may exist either because its positions are close, or because the positions in a correlated layer are close. This ambiguity, while expressive, may lead to a problem with identifiability that would be a serious issue for the interpretation of modeled social systems.

2 The Model

2.1 Problem Statement

Motivated by these problems, we propose a model with the following three objectives:

1. Use current methods from statistical SNA to model multigraphs
2. Model the layers of the multigraph in a way that can tune between comparability and expressiveness
3. Allow the model to remove redundant layers

A model with all of these qualities, which we term the Hierarchical Latent Space Model, can be described in the following way:

2.2 The Likelihood Function

Using the canonical Latent Space Model [10] as a starting point, we model a set of conditionally independent binary dyads.

The existence of each of these dyads is a function of an intercept, a set of covariates (omitted here for clarity), and the distance between the latent "position" variables z of the two nodes in the dyad. The goal of the model will be to estimate these positions. An optimal set of variables would place positions close together for nodes with an observed edge, and vice versa.

$$\eta_{ijk} = \alpha_k - \|z_{ik} - z_{jk}\|_2^2$$

The indices i and j are over the nodes; the index k refers to the different layers in the multigraph. Because the edges are binary, we use the inverse logit σ to transform η, but it should be observed that for real-valued edges, other link functions could be easily substituted.

$$\sigma(\eta_{ijk}) = \frac{1}{1 + \exp(-\eta_{ijk})} \in [0, 1]$$

All together, the unpenalized likelihood of the HLSM thus takes the form of a binomial:

$$L = \prod_k \prod_i \prod_{j<i} \sigma(\eta_{ijk})^{y_{ijk}} (1 - \sigma(\eta_{ijk}))^{1-y_{ijk}}$$

and the log likelihood is

$$\ell = \sum_k \sum_i \sum_{j<i} (y_{ijk} - 1)\eta_{ijk} - \ln(1 + \exp(-\eta_{ijk}))$$

2.3 Regularization

While the notation z for the latent variables is consistent with the literature, and somewhat more intuitive, our contribution comes from a reparameterization of the model, where

$$z_{ik} = b_i + \epsilon_{ik}$$

Our model starts with a "base" position b_i for each node, which is the hierarchical layer in the model. It then adds a layer-specific perturbation ϵ_{ik}. The behavior of the model then depends entirely on the regularization placed on the ϵ's. When there is none, each layer is fit independently. When there is an arbitrarily large amount, the perturbations are all driven to zero, and all k layers are represented identically by the base parameters b. With intermediate values, the user can find the point between these two extremes that allows for the graphs to be distinct, but also renders them sufficiently similar to be comparable, by not allowing the perturbations to stray too far from their shared base position.

Furthermore, with the use of a group lasso penalty that groups together all of the perturbations within a single layer k, the model will perform graph-wise dimensionality reduction, by setting a redundant layer equal to the base layer. This is far more valuable to the practitioner than an element-wise sparse solution, which would still require them to consider all of the layers in their multigraph.

3 Fitting

3.1 The Optimization Problem

Together these two elements form our optimization problem, which is to minimize the sum of the negative log likelihood and a penalty on the deviations ϵ_{ik}. We tested two approaches with respect to the penalty on the deviations ϵ_{ik}.

In the first model, we used an element-wise lasso, by penalizing the norm of each deviation ϵ_{ik}. This can be seen in by Equation 1. It is important to note that the latent parameters are all two-dimensional, to allow for visualization. Consequently even with this element-wise approach, the group-lasso approach and the L2 penalty is necessary, following [**?**]. However the grouping is only over the two dimensions of the individual parameter.

$$\min_{b,\epsilon} -\ell(b, \epsilon) + \lambda \sum_i \sum_k \|\epsilon_{ik}\|_2 \tag{1}$$

In the second approach, we used a group lasso penalty that is over all of the perturbations ϵ_k within each layer k, instead of on each individual deviation ϵ_{ik}. This encourages sparsity over the layers, meaning an entire graph might be zeroed out (expressed as just the base layer). This approach is illustrated in Equation 2.

$$\min_{b,\epsilon} -\ell(b, \epsilon) + \lambda \sum_k \|\epsilon_k\|_2 \tag{2}$$

Because neither of these objective functions is differentiable, we use proximal gradient descent in fitting the model.

3.2 Derivation of the Gradients

Then, the gradients of the η_{ijk} are:

$$\frac{\partial \eta_{ijk}}{\partial b_i} = -2(b_i + \epsilon_{ik} - b_j - \epsilon_{jk})$$

$$\frac{\partial \eta_{ijk}}{\partial b_j} = 2(b_i + \epsilon_{ik} - b_j - \epsilon_{jk})$$

$$\frac{\partial \eta_{ijk}}{\partial \epsilon_{ik}} = -2(b_i + \epsilon_{ik} - b_j - \epsilon_{jk})$$

$$\frac{\partial \eta_{ijk}}{\partial \epsilon_{jk}} = 2(b_i + \epsilon_{ik} - b_j - \epsilon_{jk})$$

$$\begin{aligned}
\frac{\partial \ell}{\partial b_m} &= \sum_k \sum_i \sum_{j<i} (y_{ijk} - 1)\frac{\partial \eta_{ijk}}{\partial b_m} - \frac{\partial \ln(1 + \exp(-\eta_{ijk}))}{\partial b_m} \\
&= \sum_k \left[\sum_{j<m} (y_{mjk} - 1)\frac{\partial \eta_{mjk}}{\partial b_m} - \frac{\partial \ln(1 + \exp(-\eta_{mjk}))}{\partial b_m} + \sum_{i>m} (y_{imk} - 1)\frac{\partial \eta_{imk}}{\partial b_m} - \frac{\partial \ln(1 + \exp(-\eta_{imk}))}{\partial b_m} \right] \\
&= \sum_k \Bigg[-2 \sum_{j<m} \left[(y_{mjk} - 1)(b_m + \epsilon_{mk} - b_j + \epsilon_{jk}) + \frac{\exp(-\eta_{mjk})}{1 + \exp(-\eta_{mjk})}(b_m + \epsilon_{mk} - b_j - \epsilon_{jk}) \right] \\
&\qquad + 2 \sum_{i>m} \left[(y_{imk} - 1)(b_i + \epsilon_{ik} - b_m - \epsilon_{mk}) + \frac{\exp(-\eta_{ijk})}{1 + \exp(-\eta_{mjk})}(b_i + \epsilon_{ik} - b_m - \epsilon_{mk}) \right] \Bigg] \\
&= 2 \sum_k \Bigg[-\sum_{j<m} \left(y_{mjk} - 1 + \frac{\exp(-\eta_{mjk})}{1 + \exp(-\eta_{mjk})} \right) (z_{mk} - z_{jk}) \\
&\qquad + \sum_{i>m} \left(y_{imk} - 1 + \frac{\exp(-\eta_{ijk})}{1 + \exp(-\eta_{mjk})} \right) (z_{ik} - z_{mk}) \Bigg]
\end{aligned}$$

$$\begin{aligned}
\frac{\partial \ell}{\partial \epsilon_{mk}} = 2 \Bigg[&-\sum_{j<m} \left(y_{mjk} - 1 + \frac{\exp(-\eta_{mjk})}{1 + \exp(-\eta_{mjk})} \right) (z_{mk} - z_{jk}) \\
&+ \sum_{i>m} \left(y_{imk} - 1 + \frac{\exp(-\eta_{ijk})}{1 + \exp(-\eta_{mjk})} \right) (z_{ik} - z_{mk}) \Bigg]
\end{aligned}$$

3.3 The Proximal Operator

The proximal operator for the penalty term using the L2 norm is:

$$\text{prox}_{\|.\|,\lambda t}(\epsilon_k) = \begin{cases} \frac{\|\epsilon_k\| - \lambda t}{\|\epsilon_k\|}\epsilon_k, & \|\epsilon_k\| \geq \lambda t \\ 0, & \|\epsilon_k\| < \lambda t \end{cases}$$

$$\text{prox}_{\|.\|,\lambda t}(b_i) = b_i$$

Therefore, let β be the vector with all parameters ϵ and b. The Proximal Gradient Descent method in this case is:

$$\beta^+ = \text{prox}_{\|\cdot\|,\lambda t}(\beta - t\nabla \ell(\beta))$$

3.4 Non-Convexity and Initialization

The original Latent Space Model is known to be non-convex in the latent positions z [10], and our model is no different. This issue is typically approached by using MCMC samplers [10**?**] which are capable of exploring non-convex spaces, however using them on this problem proved not only to take far more time, but also to have substantially worse results. We focus our discussion on the convex approaches used.

As we use methods designed for convex problems, we start by using a careful initialization of our procedure. This attempts to begin the algorithm close to a global minimum of the objective, so that when we do descend the likelihood, we reduce our chances of being caught in a poor local minimum. This initialization was composed of several steps.

1. Create a proposal 'base' graph from $\hat{y}_{ij} = \sum_k y_{ijk} > 0$
2. Fit a separate single-graph latent space model to each layer (including the base layer)
3. Transform the layer estimates to minimize their distance from the base estimate

The third step in this procedure was initially done with a Procrustes transform [10], as is used in Bayesian estimation of Latent Space Models. This transform is used because proposal graphs in a sampling procedure can produce a graph with many different unimportant variations, as the likelihood is invariant to translation, rotation, scaling, and flipping. The Procrustes transform is thus used to remove these transformations, while keeping the variations that are of importance to the likelihood. However this transform minimizes the overall distance between *all* pairs of points in two shapes, and in this problem, we require something slightly different. The goal here is to minimize only distances between positions that correspond to the same node – this is the property that minimizes the ϵ's, and makes the embeddings comparable.

To solve this problem we propose the Anticrustean transformation. A closed form solution, should one exist, was not derived, but a simple implementation combines line searches over the several classes of transforms to which the likelihood is invariant. Together, these steps produce a good initial estimate of all of the models parameters, which are used as starting points in the proximal gradient procedure.

4 Results

To test our model, we first simulate a toy data set consisting of a multigraph with $N = 20$ vertices and $K = 2$ layers, followed by the real and novel data set that motivated this technique. We initialize our optimization problem using the initialization procedure described in the previous section. We then run 3000 iterations of the proximal gradient algorithm. Figures 1 and 2 show the improvement in the objective function value over the 3000 iterations for, respectively, the element-wise lasso and the layer-wise lasso approaches using different values for λ. We can observe that the function values appear to converge to a local minimum. In both approaches we used a fixed step size $t = 10^{-3}$ in the proximal gradient algorithms.

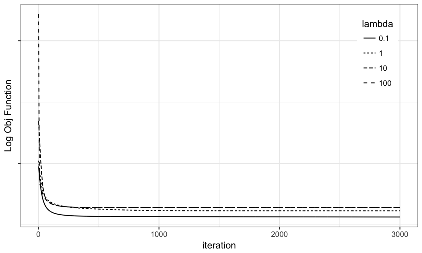

Figure 1: Value of the objective function for different values of λ in the standard lasso approach. We ran 3000 iterations of the proximal gradient descent algorithm.

Figures 3 and 4 compare the optimal positions found by each of the models with the initial one. Black points represent the base positions b_i and the red and green points (and lines) represent the two different layers of the graph. We can observe that as we increase the value of λ deviations

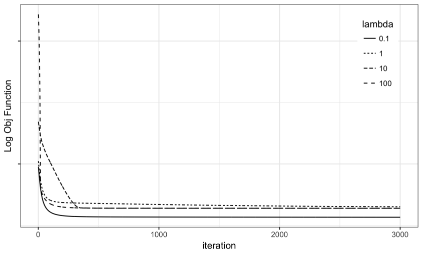

Figure 2: Value of the objective function for different values of λ in the group lasso approach. We ran 3000 iterations of the proximal gradient descent algorithm.

are more penalized and the optimal solution tends to be one where all layers collapse to a single graph. Additionally, we can also compare the resulting optimal positions between the two approached (standard lasso and group lasso). For $\lambda = 1$ the two layers are significantly more distinct than the ones in the standard lasso case.

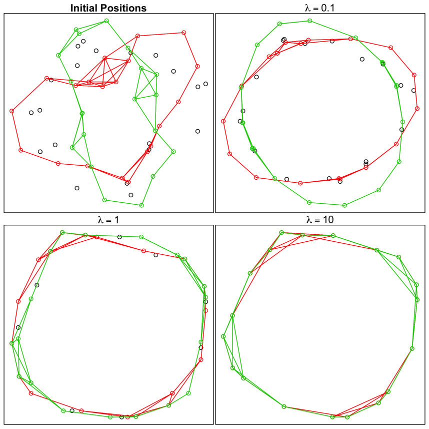

Figure 3: Scatter plots showing the initial positions (top left plot) used in the optimization problem and the final ones for different values of λ in the standard lasso approach. Black points represent the base positions b_i and the red and green points (and lines) represent the two different layers of the graph. As the value of λ increases deviations are more penalized and the optimal solution tends to be one where all layers collapse to a single graph.

In Figure 5, we plot the results of our algorithm on a novel data set of teachers at a school in West Baltimore. As can be seen, the graphs are highly visually comparable, while also maintaining a substantial amount of expressiveness to demonstrate their individual structure. We believe this convincingly demonstrates the potential utility of our technique as a tool in social network analysis, as well as many other fields.

5 Conclusion

Despite not being a convex problem, a combination of convex methods and a good heuristic starting point have been shown to efficiently fit this novel class of models. In addition, they have demonstrated all of the properties for which they were intended, and several others additionally. As demonstrated, HLSMs are capable of producing comparable graph layouts for the layers of a multigraph, and are also capable of performing layer-wise dimensionality reduction in order to meaningfully simplify this complex form of data.

HLSMs are also valuable for refining the layouts determined by current Latent Space Model implementations, as can been seen in the differences between the initializations, and the final estimates. While it is possible that many likelihoods accurately "represent" the graph (meaning essentially that their classification accuracy for edges is high [10]) , this further convex optimization is valuable

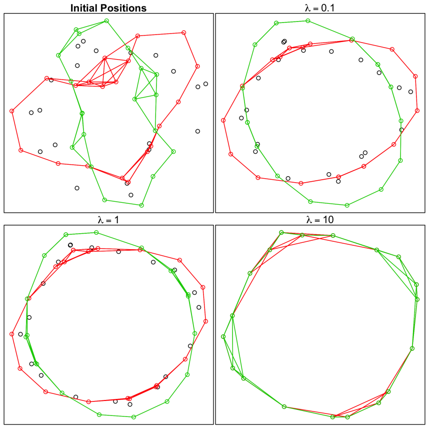

Figure 4: Scatter plots showing the initial positions (top left plot) used in the optimization problem and the final ones for different values of λ in the group lasso approach. Black points represent the base positions b_i and the red and green points (and lines) represent the two different layers of the graph. As the value of λ increases deviations are more penalized and the optimal solution tends to be one where all layers collapse to a single graph. We can observe that for $\lambda = 1$ the two layers are significantly more distinct than the ones in the standard lasso case.

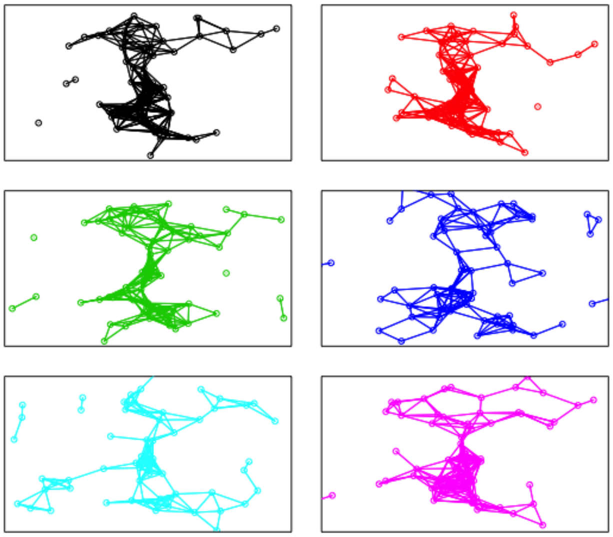

Figure 5: A Real-World Network of Teachers in West Baltimore

in finding solutions that are meaningful to human interpreters. Furthermore, our fitting procedure was found to converge to a good solution far faster than the HMC implementation with which we compared it.

Having demonstrated the basic feasibility, effectiveness, and utility, of HLSMs, the next steps will be to push its limits on different kinds of graphs, and different numbers of layers. Several different guests at our poster session mentioned additional problems, including in medical domains, and the interpretability of deep neural nets, to which our method might be able to contribute. We look forward to exploring these additional problem spaces.

NEXT:

6 Comparing Fitting Methods

Because why not (not crucial, but maybe important since it's a convex method for a non-convex problem)

6.1 Stan/HMC

The way LSMs are fit now – appear to be way slower, and worse

6.2 Coordinate-Wise Optimization

A cute idea, but one that I haven't gotten to work yet. If it comes through.

7 Visual Comparability Tests

The most important, and the most subjective. Anecdotal results above were strong though, maybe clear enough to be easy. The comparison, in every case, will be with sets of independently fit LSMs. Which though, already, given the two ovals model, we've beaten substantially, in terms of polish at least.

8 Dimensionality Reduction Tests

Create graphs of different degrees of correlation – what's the rxp between lambda, rho, and layer=0?

9 Comparability and Accuracy

What happens to the accuracy when you force a set of dissimilar graphs to be similar in latent position?

10 A Real-World Test Case

My school data, where background knowledge of the system allows validity checks for the produced layouts

10.1 Grade Clusters

Do the forced-similar graphs preserve grade clusters?

10.2 Subject Clusters

Same, for math/science/lit clusters

10.3 Role Clusters

Same, for teacher/admin/staff clusters

Else? Maybe a demonstration with another data set, like Samson? Would be cool to say something new about something so canonical.

References

[1] J Amer. Latent space models for multiview network data. 11(3):1217–1244, 2017. doi: 10.1214/16-AOAS955.

[2] Jerome Friedman and Trevor Hastie. Regularization Paths for Generalized Linear Models via Coordinate Descent. pages 1–22, 2008.

[3] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. A note on the group lasso and a sparse group lasso arXiv : 1001 . 0736v1 [math . ST] 5 Jan 2010. pages 1–8, 2010.

[4] Isabella Gollini, Thomas Brendan Murphy, Isabella G Ollini, and Thomas Brendan M Urphy. Joint Modeling of Multiple Network Views Joint Modeling of Multiple Network Views. 8600 (November), 2017. doi: 10.1080/10618600.2014.978006.

[5] Zhiwei Tony Qin and Katya Scheinberg. Efficient Block-coordinate Descent Algorithms for the Group Lasso. pages 1–23.

[6] Michael Salter-townshend and Tyler H Mccormick. Latent Space Models for Multiview Network Data. (622):1–30, 2013.

[7] Martin Vincent and Niels Richard Hansen. Sparse group lasso and high dimensional multinomial classification. pages 1–31.

[8] Tong Tong Wu and Kenneth Lange. Coordinate descent algorithms for lasso penalized regression. 2(1):224–244, 2008. doi: 10.1214/07-AOAS147.

[9] Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

[10] Peter D Hoff, Adrian E Raftery, and Mark S Handcock. Latent Space Approaches to Social Network Analysis. *Journal of the American Statistical Association*, 97(460):1090–1098, 2002. ISSN 0162-1459. doi: 10.1198/016214502388618906.

[11] Bob Carpenter, Andrew Gelman, Matthew D Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus A Brubaker, Peter Li, and Allen Riddell. Stan : A Probabilistic Programming Language. 76(1), 2017. doi: 10.18637/jss.v076.i01.

Appendix: Link to R scripts

The proximal descent methods described in this report were implemented using the R language. All scripts are available in the following git hub repository: `https://github.com/amloewi/hlsm`. The main R script is located at `https://github.com/amloewi/hlsm/blob/master/R/final_project.R`