The Latent Space Model*

University of Konstanz
Department of Computer & Information Science
Seminar Network Models, Winter Term 2009/2010
Prof. Dr. Ulrik Brandes
Advisor: Uwe Nagel

Matthias Fratz

April 1, 2010

Contents

*based on: 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.

1 Motivation

The latent space model specifically attempts to model two properties of directed networks frequently observed in social networks, namely transitivity and reciprocity [2]. These are understood probabilistically, that is, reciprocity describes the situation that, if an edge (u, v) exists, then with high probability, there also exists a reciprocal edge (v, u) — that is, the existence of (u, v) implies, with high probability, the existence of (v, u). Likewise, transitivity demands that if the network contains the edges (u, v) and (v, w), then with high probability it also contains an edge (u, w).

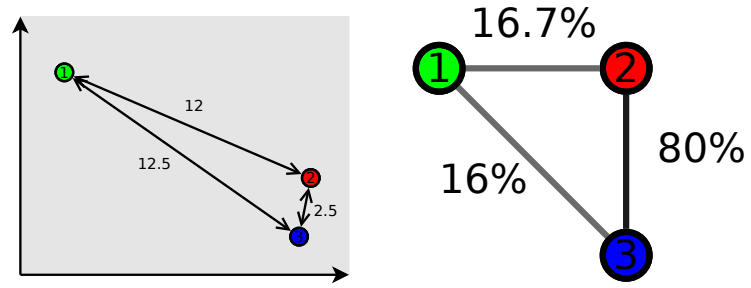
Unlike for example the Attract and Introduce model [3], which has per-node parameters controlling the indegree ("attract") and the probability of forming transitive relations ("introduce"), the latent space model models its target properties of transitivity and reciprocity indirectly as a by-product of the way edges are formed.

Another model with this property is the stochastic blockmodel [9], which assumes that the nodes belong to observed "blocks" formed by the (discrete) values of the observed characteristics. Two nodes are then considered more likely to form an edge if they are in the same block than if they are in different blocks, in addition to the per-node probabilities of forming incoming and outgoing edges. This can also be adapted to unobserved block structures [8]. In contrast to these models however, the latent space model does not divide the nodes into groups, but instead places them into an unobserved (latent) space, with the probability of an edge depending on the distance its endnodes have in the latent space.

2 The Latent Space Models

2.1 Basic Principle: The Latent Space

The latent space model associates a position z_i in the k-dimensional latent space with each node i of the network. The distance of the positions of the endnodes of an edge then controls the probability of the existence of that edge.



(a) node positions and distances in latent space

(b) example of resulting edge probabilities

Figure 1: Example of the distance of an edge's endnodes in the latent space controlling the probability that this edge is part of the network.

Assuming conditional independence, the probability of an edge (i, j) depends only on the positions z_i, z_j of its endnodes, a parameter vector Θ global to the network, and a vector x_{ij} of observed node characteristics, which in the general case is dependent on both i and j. In particular, it is assumed that the existence of an edge (i, j) is independent of the existence of any other edge (k, l):

$$\forall k, l \in V : \; k \neq i \wedge l \neq j \;\Rightarrow\; P(\mathbf{a_{ij}} = 1 | \mathbf{a_{kl}} = 1) = P(\mathbf{a_{ij}} = 1 | \mathbf{a_{kl}} = 0) \;\; ,$$

where V is the set of nodes (vertices) of the network represented by the adjacency matrix $\mathbf{A} = (\mathbf{a_{ij}})_{i,j \in V}$. Note that $\mathbf{A}$ is a random variable dependent on various model parameters.[1]

Because of the assumption of conditional independence, the probability of the network represented by a specific adjacency matrix A, given the positions $Z = (z_i)_{i \in V}$, the covariate information $X = (x_{ij})_{i,j \in V}$, and the parameters Θ, is

$$P(A | Z, X, \Theta) = \prod_{i \neq j} P(a_{ij} | z_i, z_j, x_{ij}, \Theta) \;\; . \tag{1}$$

2.2 A Generalized Model

Hoff et al. [6] present two very similar models, both of which weight the observed covariate information x_{ij} with a vector β, and use different measures of "distance" between two nodes. Furthermore, there is a "bias term" α which controls the base probability of an edge, and thus both the number of edges m and the average degree $\delta = 2\frac{m}{n}$. Using a distance function $d : V^2 \to \mathbb{R}$, the parameter vector Θ in (1) consists of α, β, and d:

$$P(A | Z, X, \alpha, \beta, d) = \prod_{i \neq j} P(a_{ij} | z_i, z_j, x_{ij}, \alpha, \beta, d) \;\; . \tag{2}$$

The probability of an edge (i, j), defined as

$$p_{ij} = P(\mathbf{a_{ij}} = 1 | z_i, z_j, x_{ij}, \alpha, \beta, d) \;\; ,$$

can then be written in a general form for both proposed models, as

$$p_{ij} = \frac{1}{1 + \frac{1}{e^{\alpha}} \cdot e^{d(z_i, z_j) - \beta \cdot x_{ij}}} \;\; . \tag{3}$$

This is the logistic function, so its logit $\eta_{ij} = \operatorname{logit} p_{ij}$ is just

$$\begin{aligned} \eta_{ij} &= \log \operatorname{odds}(a_{ij} = 1 | z_i, z_j, x_{ij}, \alpha, \beta) \\ &= \operatorname{logit} p_{ij} = \alpha + \beta \cdot x_{ij} - d(z_i, z_j) \;\; . \end{aligned} \tag{4}$$

$\beta \cdot x_{ij}$ denotes the dot product of β and x_{ij}, that is, the weighted sum of the covariate information.

[1] The notation $\mathbf{X}$ is used for a random variable named $\mathbf{X}$, while specific realizations are denoted as X. Consequently, if $\mathbf{X}$ is a matrix, its elements are termed $\mathbf{x_{ij}}$.

2.3 Geometric Distance

The most obvious choice for the distance function d is the Euclidean distance

$$d(z_i, z_j) = ||z_i - z_j|| = \sqrt{(z_i - z_j) \cdot (z_i - z_j)} \ .$$

However, any other norm could be used instead, such as one of the p-norms, for example the Manhattan norm

$$d(z_i, z_j) = ||z_i - z_j||_1 \ .$$

In the case of a p-norm, (4) would be written as

$$\eta_{ij} = \alpha + \beta \cdot x_{ij} - ||z_i - z_j||_p \ .$$

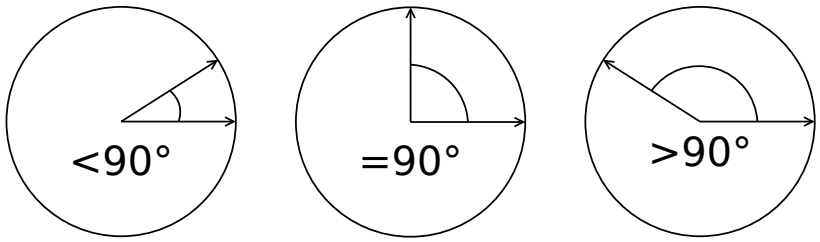
If a norm is used, the model is inherently transitive and reciprocal: If there are edges (i, j) and (j, k), they were most likely created because the nodes i and j, as well as j and k are relatively close. Due to the triangle inequality, $d(z_i, z_k)$ can be no larger than $d(z_i, z_j) + d(z_j, z_k)$, thus the probability of the transitive edge (i, k) must also be relatively high. Reciprocity is given because the norm must fulfill $d(z_i, z_j) = d(z_j, z_i)$, so the probability of reciprocal edges is equal. For "short" edges (the ones between nodes that have a small distance to each other, that is, the ones that are most likely), this translates to a high probability of the reciprocal edge.

2.4 Angular (Dis-)Similarity

Another distance metric of two nodes' position is the angular distance

$$d'(z_i, z_j) = \cos \angle(z_i, z_j) = \frac{z_i \cdot z_j}{||z_i|| \cdot ||z_j||} \ . \tag{5}$$

In this case, the "positions" of the nodes are interpreted as feature vectors representing each node's "preferences" for various characteristics. As can be seen from (5), the vectors z_k should in fact be normalized to unit length, as their length does not play any role in the equation.



(a) similar characteristics: $\cos \angle(z_i, z_j) > 0$

(b) neutral characteristics: $\cos \angle(z_i, z_j) = 0$

(c) dissimilar characteristics: $\cos \angle(z_i, z_j) < 0$

Figure 2: Graphical illustration of the $\cos \angle(z_i, z_j)$ distance metric: The more the feature vectors of two nodes "point in the same direction," (that is, the smaller the angle between them), the more similar the nodes are considered.

If all z_k are unit length, then (5) can be simplified to just the dot product:

$$d'(z_i, z_j) = \cos \angle(z_i, z_j) = z_i \cdot z_j \quad . \tag{6}$$

It might be desirable to model a node-specific degree of homophily, such that different nodes with otherwise similar characteristics can have different levels of attraction by similar and repulsion of dissimilar nodes. Doing this requires an additional parameter, because with the cosine distance metric, the "distances" would otherwise be confined to a range of ± 1 for all nodes.

This extra parameter is best added as a scaling factor of the distance, so that a homophile node "sees" larger distances to dissimilar nodes, and shorter distances to more similar nodes. Equation (6) then becomes

$$d'(z_i, z_j) = \lambda_i \cdot \cos \angle(z_i, z_j) = \lambda_i (z_i \cdot z_j) = (\lambda_i z_i) \cdot z_j \quad . \tag{7}$$

Thus it is possible to simply use the length of the feature vector z_i of the node i which is about to develop an outgoing edge, instead of actually adding another parameter.[2] By substituting $||z_i||$ for λ_i, (7) becomes

$$d'(z_i, z_j) = ||z_i|| \cdot \cos \angle(z_i, z_j) = ||z_i|| \cdot \frac{z_i \cdot z_j}{||z_i|| \cdot ||z_j||} = \frac{z_i \cdot z_j}{||z_j||} \quad . \tag{8}$$

As this is the projection of the vector z_i onto z_j, this model is called a *projection* model by Hoff et al. [6].

As can be seen from (8), the direction of the vectors encodes the feature preferences of a node, and its length encodes the node's homophily when developing outgoing edges. In cases where it is more adequate to model a node's homophily when it is about to develop an *incoming* edge, (8) can obviously be rewritten as

$$d'(z_i, z_j) = \frac{z_i \cdot z_j}{||z_i||} \quad .$$

In a situation where the product of the involved nodes' homophily is useful, (6) would be used directly.

To fit the generalized equation, the sign of the distance metric presented so far has to be flipped: The distance function d' returns larger values for more similar nodes, which is against the intuitive meaning of distances, the one used in (4). Thus:

$$d(z_i, z_j) = -d'(z_i, z_j) = -\frac{z_i \cdot z_j}{||z_j||} \quad .$$

2.5 Less Restrictive Similarity Metrics

In fact, one might want to do away with the distance function entirely, and instead use a full $n \times n$ matrix of distances between each pair of the n nodes, without enforcing the axioms required for a norm. This, however, is not very desirable — not just because of the more difficult visualization and interpretation, but because of severe overfitting concerns: For any given network with

[2] It is worth pointing out that by forcing the vector's length to be unity, one degree of freedom was removed, so in fact one parameter was unnecessary. It is this redundant parameter which is recycled. As it could have just been removed, there is in fact a new parameter that is added.

an adjacency matrix $A' = (a'_{ij})$, it is possible to define a model which fits this network arbitrarily well, but fits no other networks at all. This is done simply by setting $\alpha = \gamma$, $\beta = \mathbf{0}$ (the zero vector) and

$$d(z_i, z_j) = \begin{cases} 0 & \text{if } a'_{ij} = 1 \\ 2\gamma & \text{if } a'_{ij} = 0 \end{cases} .$$

That way, (4) becomes

$$\eta_{ij} = \alpha - d(z_i, z_j) = \begin{cases} \gamma & \text{if } a'_{ij} = 1 \\ -\gamma & \text{if } a'_{ij} = 0 \end{cases} .$$

By letting γ approach infinity, on an undirected network, one can then get the probability expressed by (3) arbitrarily close to a perfect (over-)fit:

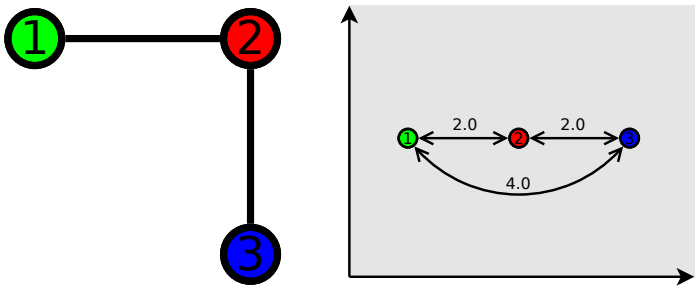
$$p_{ij} = \begin{cases} 1 & \text{if } a'_{ij} = 1 \\ 0 & \text{if } a'_{ij} = 0 \end{cases} .$$

In addition to this overfitting problem, there is also a risk of losing the two inherent properties of transitivity and reciprocity, both of which are normally guaranteed by the axioms of a norm.

3 Parameter Estimation

3.1 Parameter Estimation Basics

In general, the model parameters will not be known. While this model could be used to generate networks by choosing the parameters randomly, normally there will be a given network $A' = (a'_{ij})$ to which the model is to be fitted. In order to do this, it is assumed that the given network was generated by a latent space model using a k-dimensional space. The dimensionality k and the distance metric d must be chosen arbitrarily, but all other parameters can then be found by optimizing the model in such a way that the probability of the model generating the observed network is maximized.



(a) observed network (b) generated positions

Figure 3: Parameter estimation example. The optimization algorithm tries to find the positions that have the highest probability of generating the observed network.

Given the observed network A' and the covariate information X generated from observed characteristics, this optimization is accomplished by maximizing the probability $P(\mathbf{A} = A'|Z, X, \alpha, \beta, d)$.

3.2 Derivation of the Optimization Goal

Using the conditional independence expressed in (2), the probability $P(A = A'|Z, X, \alpha, \beta, d)$ can be written as

$$\begin{aligned} P(\mathbf{A} = A'|Z, X, \alpha, \beta, d) &= \prod_{i \neq j} P(\mathbf{a_{ij}} = a'_{ij}|z_i, z_j, x_{ij}, \alpha, \beta, d) \\ &= \prod_{i \neq j} \begin{cases} p_{ij} & a'_{ij} = 1 \\ 1 - p_{ij} & a'_{ij} = 0 \end{cases} . \end{aligned}$$

As the logarithm is a monotonic, strictly increasing function, it is possible to instead maximize the logarithm of $P(\mathbf{A} = A'|Z, X, \alpha, \beta, d)$. A much simpler optimization goal can then be derived.

$$\begin{aligned} \log P(\mathbf{A} = A'|Z, X, \alpha, \beta, d) &= \log \prod_{i \neq j} \begin{cases} p_{ij} & a'_{ij} = 1 \\ 1 - p_{ij} & a'_{ij} = 0 \end{cases} \\ &= \sum_{i \neq j} \log \begin{cases} p_{ij} & a'_{ij} = 1 \\ 1 - p_{ij} & a'_{ij} = 0 \end{cases} \\ &= \sum_{i \neq j} \log \begin{cases} \frac{1}{1+e^{-\eta_{ij}}} & a'_{ij} = 1 \\ 1 - \frac{1}{1+e^{-\eta_{ij}}} & a'_{ij} = 0 \end{cases} \end{aligned}$$

Because of $1 - \frac{1}{1+e^{-\phi}} = \frac{1+e^{-\phi}-1}{1+e^{-\phi}} = \frac{e^{-\phi}}{1+e^{-\phi}} = \frac{e^{-\phi}e^{\phi}}{(1+e^{-\phi})e^{\phi}} = \frac{1}{e^{\phi}+1}$, this can be rewritten as follows.

$$\begin{aligned} \log P(\mathbf{A} = A'|Z, X, \alpha, \beta, d) &= \sum_{i \neq j} \log \begin{cases} \frac{1}{1+e^{-\eta_{ij}}} & a'_{ij} = 1 \\ \frac{1}{1+e^{+\eta_{ij}}} & a'_{ij} = 0 \end{cases} \\ &= \sum_{i \neq j} \log \begin{cases} 1 - \frac{1}{1+e^{\eta_{ij}}} & a'_{ij} = 1 \\ \frac{1}{1+e^{\eta_{ij}}} & a'_{ij} = 0 \end{cases} \\ &= \sum_{i \neq j} \log \begin{cases} \frac{1+e^{\eta_{ij}}-1}{1+e^{\eta_{ij}}} & a'_{ij} = 1 \\ \frac{1}{1+e^{\eta_{ij}}} & a'_{ij} = 0 \end{cases} \\ &= \sum_{i \neq j} \log \begin{cases} \frac{e^{\eta_{ij}}}{1+e^{\eta_{ij}}} & a'_{ij} = 1 \\ \frac{1}{1+e^{\eta_{ij}}} & a'_{ij} = 0 \end{cases} \end{aligned}$$

Because $(e^{\eta_{ij}})^0 = 1$ and $(e^{\eta_{ij}})^1 = e^{\eta_{ij}}$, this can also be written without conditional expressions.

$$\begin{aligned} \log P(\mathbf{A} = A'|Z, X, \alpha, \beta, d) &= \sum_{i \neq j} \log \frac{(e^{\eta_{ij}})^{a'_{ij}}}{1 + e^{\eta_{ij}}} \\ &= \sum_{i \neq j} \eta_{ij} a'_{ij} - \log(1 + e^{\eta_{ij}}) \end{aligned} \tag{9}$$

3.3 Parameter Estimation Algorithm

Using a $n \times n$ matrix of distances as described in section 2.5, distances maximizing $\log P(\mathbf{A} = A'|Z, X, \alpha, \beta, d)$ are easily found with any suitable optimization algorithm. Unfortunately, these are pairwise distances that do not necessarily represent the distances in any space with a reasonable number of dimensions, such as the desired k-dimensional space. In fact, the parameter space for distances in the k-dimensional space is too complex for linear optimization, and must be approached with nonlinear optimization strategies such as genetic algorithms or Markov chains Monte Carlo methods.

One possible solution is to first determine pairwise distances between the nodes. For these distances, an approximation can then be found in the k-dimensional space by applying multidimensional scaling [4]. These initial positions are then further optimized using a Metropolis-Hastings algorithm [7].

During this last optimization step, there is a precaution that should be taken to avoid one possible problem: If the distance-based model presented in section 2.3 is used, the generated positions can be rotated, mirrored and translated arbitrarily without affecting the resulting distances, and thus, the model. Likewise, for the angular dissimilarity metric presented in section 2.4, while the space cannot be translated anymore, it can still be rotated and mirrored at will without any changes to the edge probabilities. Thus there are infinitely many sets of parameters that are equivalent with respect to the edge probabilities they generate.

To prevent the algorithm from failing to converge simply because it keeps performing these irrelevant transformations, it is therefore necessary to represent each of these equivalent classes with a unique representative. To obtain this, if the Metropolis-Hastings algorithm has just accepted a new set of parameters, then before starting the next iteration, the positions are canonicalized using a Procrustes transformation [5], so that the positions are closest to a set of defined reference positions, such as the initial positions generated using multidimensional scaling.

Hoff et al. [6] also suggest obtaining the initial set of distances using two alternative, simpler, graph-theoretical methods from Wassermann et al. [10]. The first approach interprets the rows or columns of the adjacency matrix as vectors, and calculates the distance between nodes as the Euclidean distance of their entries in the adjacency matrix, whereas the second method uses the shortest-path distance of the nodes in question, that is, the length of the shortest path between these nodes.

The algorithm can thus be summarized as follows:

1. generate initial pairwise distances between the nodes, using either
   - the distances maximizing (9)
   - ad-hoc distances between the nodes
2. convert the distances to positions Z_0 using multidimensional scaling
3. maximize (9) using a Metropolis-Hastings algorithm:
   (a) generate new candidate positions Z_{s+1}

(b) accept them with a probability of $\frac{P(Z_{s+1})}{P(Z_s)} \cdot \frac{P(A=A'|Z_{s+1},X,\alpha,\beta,d)}{P(A=A'|Z_s,X,\alpha,\beta,d)}$, otherwise let $Z_{s+1} = Z_s$

(c) if the new positions were accepted, use Procrustes transformation to move them as close as possible to Z_0

(d) update α and β by generating candidates and probabilistically accepting them

4. repeat step 3 until the algorithm converges

Instead of waiting for convergence, one may also use a predefined number of iterations, provided that this number is sufficiently large that convergence is likely reached.

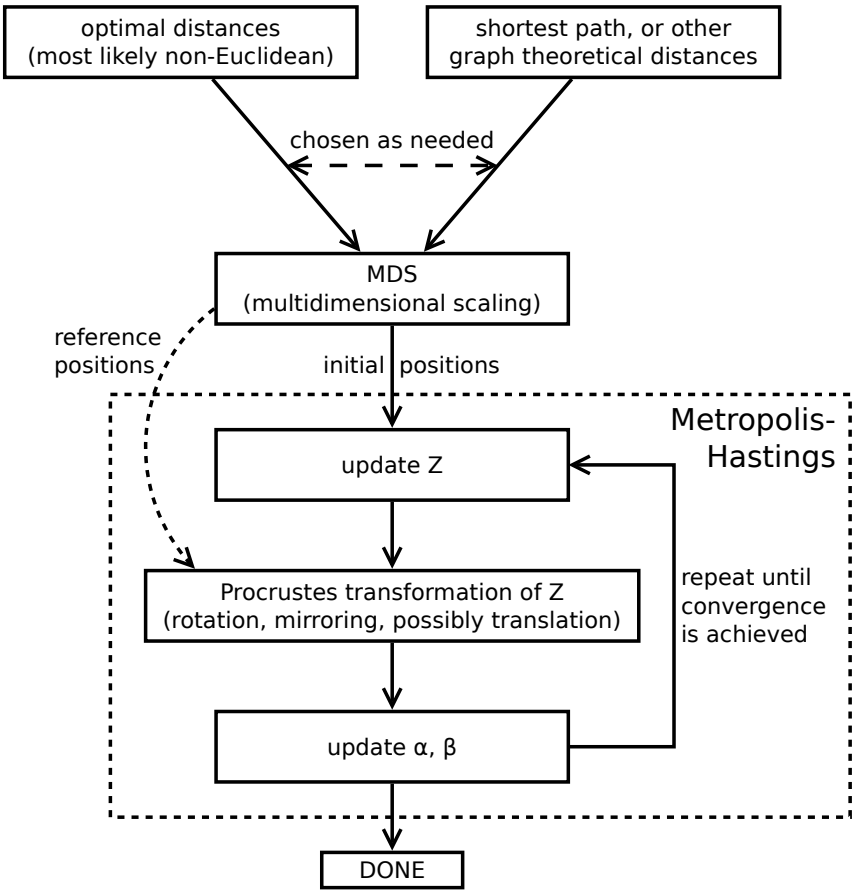


Figure 4: Flowchart of the parameter estimation algorithm. The initial distances are converted to positions using multidimensional scaling, and then optimized using an equivalence-aware Metropolis-Hastings algorithm.

4 A Related Model: The Social Space

4.1 A Feature Comparison

The latent space model is quite similar to a related, but newer, model, the Social Space Model by Boguñá et al. [1]. Both models place the nodes of the network in some unobserved space, with the distance of two nodes determining the the probability of an edge between them.

However, while with the latent space model, there is no direct relation between the dimensions of the latent space and the features of the nodes, the social space model, at least conceptually, uses one subspace for every feature of the nodes. Each such subspace features its own distance metric, which is potentially unique to this subspace only. The distance in all these subspaces are then combined to form an global distance by a weighted sum of the individual subspace distances. The latent space model, in contrast, only has a single distance metric for the entire, undivided latent space, so different weights for different dimensions — there is, in the general case, no such thing as a feature's dimension — can only be modeled by scaling that entire dimension, which, fortunately, is an unproblematic option.

The edge probability equation of the social space model, corresponding to (3) for the latent space model, is

$$p_{ij} = \sum_{n=1}^{\nu} \omega_n \frac{1}{1 + \left(\frac{d_n(z_i, z_j)}{b_n}\right)^{\alpha_n}} \quad , \tag{10}$$

where ν is the number of subspaces, each of which potentially consists of multiple dimensions; ω_n is a weight factor for this subspace's distances, that is, it controls the importance of this subspace; α_n is a homophily parameter, b_n controls the average degree and d_n is a subspace-specific distance metric.

While there is no explicit support for observed characteristics, it is of course trivial to use one or more subspaces for them, fixing the nodes' positions in this space, but not, for example, its weight scale and homophily parameter. So in fact, there is simply no distinction between observed and unobserved node features.

Last but not least, the distances in the social space model are scaled polynomially, by taking them to the α_n-th power, while the latent space model scales them exponentially as $e^{d(z_i, z_j)}$.

4.2 Similarities Between Simplified Versions

Despite the distances between the latent space model and the social space model, there is a striking similarity between the two models. This is most readily seen in a simplified version of the two models, by omitting the covariate information from the latent space model and thus assuming that nothing is known about the nodes, and, on the other hand, restricting the social space model to a single subspace ($\nu = 1$). Because this single subspace can still consist of several dimensions, restricting the social space model this way does not reduce the model to the one-dimensional case, which would have been a very unrealistic assumption.

Simplifying (10) for a single feature yields:

$$p_{ij} = \frac{1}{1 + \frac{1}{b^\alpha} \cdot d(z_i, z_j)^\alpha} \ .$$

This is already very similar to (3) of the latent space model, which, with the covariate information omitted, reads

$$p_{ij} = \frac{1}{1 + \frac{1}{e^\alpha} \cdot e^{d(z_i, z_j)}} \ .$$

The similarity can be shown even more clearly by showing the logit of these probabilities. Substituting $b' = \log b^\alpha$, the simplified social space model is described by

$$\begin{aligned} \eta_{ij} = \operatorname{logit} p_{ij} &= \alpha \log b - \alpha \log d(z_i, z_j) \\ &= b' - \alpha \log d(z_i, z_j) \ . \end{aligned} \tag{11}$$

For comparison, the corresponding logit for the latent space model without covariate information is

$$\eta_{ij} = \alpha - d(z_i, z_j) \ .$$

It can be seen that b' (and thus b) in the social space model plays the same role as α does in the latent space model, namely that of a bias term describing the base probability of edges. Nevertheless, the difference between exponential and polynomial scaling of the distances remains in the form of the logarithm in (11).

References

[1] Marián Boguñá, Romualdo Pastor-Satorras, Albert Díaz-Guilera, and Alex Arenas. Models of social networks based on social distance attachment. *P*hysical Review E, 70:056122, 2004.

[2] James A. Davis. Clustering and hierarchy in interpersonal relations: Testing two graph theoretical models on 742 sociomatrices. *A*merican Sociological Review, 35(5):843–851, 1970.

[3] James H. Fowler, Christopher T. Dawes, and Nicholas A. Christakis. Model of genetic variation in human social networks. *P*roceedings of the National Academy of Sciences of the United States of America, 106(6):1720–1724, 2009.

[4] John C. Gower. Some distance properties of latent root and vector methods used in multivariate analysis. *B*iometrika, 53(3–4):325–338, 1966.

[5] John C. Gower and Garmt B. Dijksterhuis. *P*rocrustes Problems. Oxford University Press, 2004.

[6] Peter D. Hoff, Adrian E. Raftery, and Mark S. Handcock. Latent space approaches to social network analysis. *J*ournal of the American Statistical Association, 97(460):1090–1098, 2002.

[7] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *J*ournal of Chemical Physics, 21(6):1087—-1092, 1953.

[8] Krzysztof Nowicki and Tom A. B. Snijders. Estimation and prediction for stochastic blockstructures. *J*ournal of the American Statistical Association, 96(455):1077–1087, 2001.

[9] Yuchung J. Wang and George Y. Wong. Stochastic blockmodels for directed graphs. *J*ournal of the American Statistical Association, 82(397):8–19, 1987.

[10] Stanley Wasserman and Katherine Faust. *S*ocial Network Analysis: Methods and Applications. Cambridge University Press, 1994.