A Latent Space Mapping for Link Prediction

Anthony Brew, Michael Salter-Townshend
Clique
University College Dublin
{anthony.brew,michael.salter-townshend}@ucd.ie

Abstract

Network modeling can be approached using either discriminative or probabilistic models. In the task of link prediction a probabilistic model will give a probability for the existence of a link; while in some scenarios this may be beneficial, in others a hard discriminative boundary needs to be set. Hence the use of a discriminative classifier is preferable. In domains such as image analysis and speaker recognition, probabilistic models have been used as a mechanism from which features can be extracted. This paper examines using a probabilistic model built on the entire graph to extract features to predict the existence of unknown links between two nodes. It demonstrates how features extracted from the model as well as the predicted probability of a link existing can aid the classification process.

1 Introduction

Link prediction is the problem of detecting the existence of a link between two items based on their attributes and the existence of other observed links [6].

This paper is concerned with making predictions on links in network datasets. Many network datasets include links, non-links and unobserved links; e.g. in mobile phone network data each carrier knows whether there exists a link between two customers on their own network but cannot determine whether two off-carrier nodes are linked or not. Other uses of link prediction include identification of links (or non-links) that are surprising under a fitted model or prediction of future link patterns given the current network.

In this paper we investigate exploiting the properties of a globally built probabilistic model to aid in the classification of missing links via a discriminative classifier. This type of strategy has been employed successfully in other domains such as speaker recognition and image analysis, for example the Fisher kernel [9].

Probabilistic models focus their efforts on producing a predictive probability for a given classification task whereas discriminative classifiers focus their attention on finding a direct mapping from the inputs $\mathbf{x}$ to the labels y [11]. However in the process of building a generative model a wealth of information is learnt about the global problem that can be exploited to aid in the decision function of the chosen discriminative classifier.

In the most basic case the probability of a link found from the probabilistic model needs to be thresholded to give a clear decision boundary. Depending on the ratio of links to non-links and the fit of the model to the data, transitivity, etc, this may be quite different from 0.5. In speaker verification, Bengio and Mariéthoz suggested that likelihoods coming from the generative models classically used in this domain may not be perfect and that this may be corrected by post-processing the scores through a discriminative classifier to make decisions [2]. Other mapping such as the Fisher kernel go one step further by extracting features from the underlying generative model to map the item to be classified into a new feature space. In essence the global model becomes the object from which features are extracted.

In this paper we investigate extracting features from a globally built probabilistic model, namely the Latent Position Cluster Model (LPCM) [7] and passing these features to a discriminative classifier, namely a Support Vector Machine. The paper proceeds as follows: In Section 2 an overview of link prediction using generative models is described. In Section 2.1 the Latent Position Cluster Model is described. In Section 2.2 the features that are extracted are described. In Section 2.3 the Support Vector Machine is introduced. Some experimental results are then given in Section 3 and conclusions are drawn in 4.

2 Link Prediction Based on Probabalistic Models

Link prediction of unobserved links in a network, given the observed links and non-links is a problem that has received growing attention; see for example [10], [4] and [15]. However only [5] makes use of probabilistic models for such data. The focus in that paper is on taking a deterministic model for binary data and using it to construct a probabilistic model.

Raftery et al. includes link prediction under one type of fully probabilistic model, namely the Latent Position Cluster Model [12]. This model also forms the basis of the work we present here; our contribution is to add a deterministic classifier to the analysis. The prediction of links using such a Bayesian probabilistic model reduces to thresholding the posterior predictive probability of a link. If the network is d_k representable for some dimension k (see [8]) then there is a clear linear decision boundary that will perfectly separate the links and non-links; in the more common situation where the network is not d_k representable, we must use some other method for choosing the threshold of probability above which we predict a link. This will not necessarily be 0.5, depending on the fit of the model to the data, ratio of links to non-links, etc.

We begin by demonstrating the training of a linear classifier on these probabilities and then extend the method to incorporate other features of the fitted probabilistic model into the classifier. Specifically, we include all node-pair specific terms appearing in the posterior expectation of the log-likelihood of the each potential link along with the first derivatives of the expected log-likelihood w.r.t. each of these terms. These derivatives are calculated within the framework of fitting the Bayesian probabilistic model for the network using Variational Bayes.

To our knowledge, such a method has not been attempted elsewhere. We demonstrate in the results section that a performance boost in link prediction is achieved by incorporating these additional features in the classifier. This is preliminary work and we expect a further performance boost is achievable. This section proceeds by introducing the Latent Position Cluster Model 2.1, how the model is fitted 2.1.1, the feature mapping induced from the model 2.2 and finally a brief introduction to Support Vector Machines 2.3.

2.1 Latent Position Cluster Model

Some recent approaches to modeling social networks have focussed on embedding the actors in a latent “social space” (see [8]). Under such models, links are more likely for actors that are close in social space than for actors that are distant. This approach models the transitivity common to many network datasets. Homophily by attributes is also readily obtainable via the inclusion of covariate information.

In addition, the Latent Position Cluster Model (LPCM) of [7] explicitly models the clustering that is exhibited in many network datasets. [13] demonstrated inference on such models using Variational Bayes, in a similar spirit to the inference used in [1] on Mixed-Membership Stochastic Blockmodels (MMSBs).

In the LPCM, a binary interactions data matrix Y is modelled using logistic regression in which the probability of a link between two nodes depends on the distance between the nodes in the latent space:

$$\text{log-odds}(y_{a,b} = 1 | z_a, z_b, \beta) = \beta - |z_a - z_b| \tag{1}$$

where β is an intercept parameter and $|z_a - z_b|$ is the Euclidean distance between the latent positions Z of nodes i and j. In addition, the links are assumed to be independent conditional on the latent positions of the actors in the latent space.

To represent the clustering, the latent positions Z are modeled as coming from a mixture of G multivariate normal distributions:

$$z_a \sim \sum_{g=1}^{G} \lambda_g \mathrm{MVN}_d(\mu_g, \sigma_g^2 I_d) \tag{2}$$

where λ_g is the probability that actor i belongs to the g^{th} group, so that $\lambda_g \geq 0, \{g = 1, \ldots G\}$ and $\sum_{g=1}^{G} \lambda_g = 1$, and I_d is the $d \times d$ identity matrix.

The parameters β, λ, σ and μ are given hyper-prior distributions.

2.1.1 Variational Bayesian Inference

We fit the above probabilistic model for the network data as per [13]. The variational approximation to the posterior is a fully factorised product (mean field approximation). Focusing on the terms appearing in Equation (1), the variational posterior for the latent positions z is given by $q_z = \mathrm{MVN}_d(\tilde{z}, \tilde{\sigma}^2)$ where d is the dimension of the latent space. The variational posterior for β is given by $q_\beta = \mathrm{Normal}(\tilde{\xi}, \tilde{\psi}^2)$ We use the tilde over the terms to indicate a variational parameter.

The posterior expectation of the log-likelihood using the variational approximation was given in [13] by

$$\begin{aligned} P_{a,b} = \mathbf{E}_q[\log(p(y_{a,b}|\beta, Z))] \quad &= \quad \tilde{\xi} - \left(|\tilde{z}_a - \tilde{z}_b|^2 + d(\tilde{\sigma}_a^2 + \tilde{\sigma}_b^2)\right)^{\frac{1}{2}} \\ &- \quad \log\left(1 + \exp\left(\tilde{\xi} - \left(|\tilde{z}_a - \tilde{z}_b|^2 + d(\tilde{\sigma}_a^2 + \tilde{\sigma}_b^2)\right)^{\frac{1}{2}} + \frac{\tilde{\psi}^2}{2}\right)\right) \end{aligned} \tag{3}$$

Note that only the intercept terms $\tilde{\xi}$ and $\tilde{\psi}^2$ and the positional terms $\tilde{z}_a, \tilde{z}_b, \tilde{\sigma}_a^2$ and $\tilde{\sigma}_b^2$ appear in the expected log-likelihood.

2.2 The Mapping

The LPCM model provides a posterior predictive probability $P_{a,b}$ of a link for each node pair a, b going from node a to node b. The first mapping that can be used to train a classifier is simply the one dimensional vector $P_{a,b}$ for each node pair.

In order to create a more elaborate mapping which depends on information learned by the LPCM model we include the variational parameters as features:

$$\tilde{v}_{a,b} = \left\{\tilde{z}_a, \tilde{z}_b, \tilde{\sigma}_a^2, \tilde{\sigma}_b^2, \frac{\partial P_{a,b}}{\partial \tilde{z}_a}, \frac{\partial P_{a,b}}{\partial \tilde{z}_b}, \frac{\partial P_{a,b}}{\partial \tilde{\sigma}_a^2}, \frac{\partial P_{a,b}}{\partial \tilde{\sigma}_b^2}\right\}^T$$

Namely this is made up by the positions of each node in the latent space $\tilde{z}_a$ and $\tilde{z}_b$ determined by LPCM model, the models uncertainty in their positions in that space $\tilde{\sigma}_a^2$ and $\tilde{\sigma}_a^2$ and the first order partial derivatives of the expected log-likelihood w.r.t. these.

2.3 Support Vector Machines

In this work we have employed Support Vector Machines (SVM's) as the discriminative classifier [14] used to learn a decision function from the features extracted from the probabilistic LPCM model. SVM's have been applied to many problems in classification and regression tasks, generally yielding good performance compared with other algorithms. The resulting classifier is of the form

$$y_{a,b} = \sum_{j=1}^{l} y_j \alpha_j K(\mathbf{x_{a,b}}, \mathbf{x_j}) + d$$

where $\mathbf{x}_j$ is a node pair represented as a set of features extracted from the probabilistic LPCM model, $y_j = 1$ if the pair $\mathbf{x}_j$ is connected by an edge and $y_j = -1$ if the pair $\mathbf{x}_j$ is known not to be an edge[1]. l is the number of training pairs, α_j and d are parameters of the trained SVM model and $K(\mathbf{x_{a,b}}, \mathbf{x_j})$ is a kernel function that can have different forms, the simplest being

$$K(\mathbf{x_{a,b}}, \mathbf{x_j}) = \mathbf{x_{a,b}}\mathbf{x_j}$$

which leads to a linear SVM, a more general kernel is the Radial Basis Function (RBF) kernel

$$K(\mathbf{x_{a,b}}, \mathbf{x_j}) = \exp\left(\frac{-||\mathbf{x_{a,b}} - \mathbf{x_j}||^2}{w^2}\right)$$

where w is the "kernel width" a parameter which needs to be selected (via cross validation for instance) and determines how flexible the resulting decision surface will be.

It is important to note that in the normal SVM formation the training criterion aims to minimise the number of classification errors. For this reason in this preliminary work our primary evaluation metric has been accuracy. A good introduction to SVM's can be found in [3].

3 Evaluation

This section proceeds by describing the data and the experimental setup used in the evaluation, followed by a discussion on the results found.

3.1 Datasets

The evaluation of the feature space mapped by the LPCM model described in Section 2.2 was performed on three datasets; *Simulated*, *Facebook* and *S50*. These datasets are described as follows:

For the simulated dataset the binary interactions matrix Y was generated in the following manner, for 6 groups in a 2d Euclidean social space:

$$\begin{array}{rcl} \beta & = & 1.0 \\ \mu & = & \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \\ \mu_5 \\ \mu_6 \end{bmatrix} = \begin{bmatrix} -9 & -6 \\ -3 & 0 \\ -6 & 3 \\ 0 & -6 \\ -6 & -3 \\ 0 & 3 \end{bmatrix} \\ \sigma^2 & = & = [\sigma_1^2, \sigma_2^2, \sigma_3^2, \sigma_4^2, \sigma_5^2, \sigma_6^2] = [0.17, 0.11, 0.15, 0.11, 0.12, 0.14] \\ z_i & \sim & \text{MVN}_2(\sum K_{ig}\mu_g, \sum K_{ig}\sigma_g^2 I_2) \\ Y_{i,j} & \sim & \text{Bernoulli}(\text{logit}^{-1}(\beta - |z_i - z_j|)) \end{array}$$

for $i, j = 1, \ldots, N$ and with K constructed such that the split of the actors across the groups is $\{7, 7, 8, 10, 11, 17\}$. We then replace 10% of all possible links with unknown links in a symmetric fashion (i.e. if link $\{a, b\}$ is broken then so is $\{b, a\}$). Both links and non-links are replaced with unobserved values. The model must therefore be fitted on an incomplete dataset and so will not be a perfect fit to the data.

We also examine an egocentric network in which the actors are members of the social networking website Facebook. The egocentric network for a single actor (an author of this paper) comprises the 81 "Facebook friends" with which the actor is linked. We remove this central actor and explore the structure of the links between the remaining nodes. The data was generated using a Facebook application that is available at `http://learning101.posterous.com/bernie-hogan-facebook-friend-data`

Lastly, we looked at the public dataset S50 excerpt of "Teenage Friends and Lifestyle Study" available at `http://www.stats.ox.ac.uk/~snijders/siena/s50_data.htm` We examine the aggregated-over-time friendship network of the largest connected component (48 actors). We did not include the covariate information available in the dataset in this study.

[1] Where links are unknown in the training set they are unused

3.2 Experimental Setup

The method of creating a projection based on the global LPCM model was evaluated as follows: to reduce the overall run time for the experiment, global parameters for the LPCM model were selected. The parameters that were tuned using Bayes information criterion were the dimension of the latent space and the number of communities.

The experiment was then run 1000 times and during each run 10 links (symmetrically and not including self links) were randomly set to unknown and their true labels were held back for testing, (i.e $10,000$ tests are performed per dataset). The LPCM model was then built using the globally selected parameters. Each set of features ($P_{a,b}$ only, $P_{a,b}$ with variational parameters $\tilde{v}_{a,b}$) as described in Section 2.2 were then extracted from the model and used to train an SVM on link and non-link data using parameters selected via cross-validation in the first run. Support Vector Machines are not invariant to linear transformations and so the feature vectors were preprocessed to have unit variance and zero mean in the training set. Self links were assumed to exist and were used during training however the resulting classifier was not tested using self links as they were considered trivial to classify.

When probability $P_{a,b}$ alone was selected to be used as a feature a linear SVM was used to select a threshold that maximised the margin between the links and non-links, the C parameter of the SVM was selected from the set $\{2 \times 10^5, 2 \times 10^3, 2 \times 10^1, 2 \times 10^{-1}, 2 \times 10^{-3}, 2 \times 10^{-5}\}$. There is no reason to believe that the full feature set $\{P_{a,b}, \tilde{v}_{a,b}\}$ described in Section 2.2 is linearly separable and so a RBF kernel was used. The parameters for the RBF kernel were selected via Grid search from the same set of possible C values used in the case of the linear classifier and the kernel width w was selected from the set $\{200.0, 100.0, 50.0, 20.0, 10.0, 5.0, 2.0, 1.0, 0.5, 0.2\}$

3.3 Results

Table 1 shows the results found for each of the datasets. The primary evaluation metric is accuracy for reasons explained in section 2.3. Due to the heavy imbalance in the dataset balanced accuracy [2] is included to help tease out further the performance differences found. Row one shows the performance of the SVM when a linear kernel was used and only the probability $P_{a,b}$ of a link existing was provided to the classifier. Row two is mainly included for completeness, this shows the performance of an SVM when a RBF kernel was employed again using only $P_{a,b}$ as the only feature. Row three demonstrates our main finding where the SVM classifier is trained using a RBF kernel utilising both the probability of a link existing $P_{a,b}$ and the variational parameters $\tilde{v}_{a,b}$ extracted from the LPCM model.

It was observed for both the Facebook and Simulated data, that accuracy increased when variation features $\tilde{v}_{a,b}$ were included as features in training the classifier.

Performance was not found to improve for the S50 dataset. When the breakdown in accuracy was examined more closely it was found that the classifier had degenerated into a ‘classify all’ classifier predicting most pairs as “non-link”. Note that the distribution of the predictive probabilities for the unknown links upon which we evaluate our methods can be markedly different from the distributions for both the links and non-links sets. We believe this is due to the LPCM model providing a poor fit for this dataset and so a poor “object” for extracting features from for the post-processing discriminative classifier. This result highlights that it is unrealistic to believe a classifier is being trained on features from a poorly fitting model can magically fix the overly poor fit of the probabilistic model. This suggests post processing by a classifier can apply make-up to a well shaped face but there’s no point putting lipstick on a pig.

To examine further the performance increase on the Facebook and Simulated dataset we include confusion tables found across the 10,000 tests performed in Tables 2 and 3. It is clear when only $P_{a,b}$ is used as a feature, the main source of error in the classification is the false-negative rate. By including the variational parameters $\tilde{v}_{a,b}$ an increase of more than 10% in the number of links detected is observed.

[2] Balanced Accuracy: the macro-average of the accuracy for each class

Method	Accuracy			Balanced Accuracy		
	Facebook	Simulated	S50	Facebook	Simulated	S50
Linear SVM $\{P_{a,b}\}$	89.8%	90.6%	*90.2%*	73.4	75.7	*56.8*
RBF SVM $\{P_{a,b}\}$	89.8%	90.1%	*90.6%*	73.3	72.8	*55.5*
RBF SVM $\{P_{a,b}, \tilde{v}_{a,b}\}$	**91.5%**	**91.3%**	*90.3%*	**79.1**	**79.3**	*56.5*

Table 1: Accuracy and balanced accuracy for classification using LPCM's output probability $P_{a,b}$ and variational parameters $\tilde{v}_{a,b}$ to predict links

Table 2: Facebook Confusion Tables

(a) Linear SVM $\{P_{a,b}\}$

Real \Pred	Link	Non-Link	#
Link	50.3%	49.7%	1430
Non-Link	3.6%	96.4%	8570

(b) RBF SVM $\{P_{a,b}, \tilde{v}_{a,b}\}$

Real \Pred	Link	Non-Link	#
Link	61.8%	38.2%	1430
Non-Link	3.6%	96.4%	8570

Table 3: Simulated Data Confusion Tables

(a) Linear SVM $\{P_{a,b}\}$

Real \Pred	Link	Non-Link	#
Link	54.6%	45.4%	1462
Non-Link	3.2%	96.8%	8538

(b) RBF SVM $\{P_{a,b}, \tilde{v}_{a,b}\}$

Real \Pred	Link	Non-Link	#
Link	62.4%	37.6%	1462
Non-Link	3.8%	96.2%	8538

4 Conclusions

This work has demonstrated how using the probability $P_{a,b}$ of a link existing between two nodes derived from a probabilistic model to learn a threshold for discriminative classification can be substantially improved by providing the other features based on the probabilistic model. This is provided that the underlying probabilistic model is a reasonable fit for the data to begin with. The approach outlined in this paper should in principle work for any probabilistic model coupled with any choice of post-processing classifier.

In this preliminary work a vanilla SVM was applied whose objective function aims to minimise classification errors. In link prediction it is not generally the case that one wishes to minimise classification error but other metrics. One approach that merits further investigation is to employ a weighted classifier to minimise other types of error in the objective function of the chosen classifier.

References

[1] E.M. Airoldi, D.M. Blei, S.E. Fienberg, and E.P. Xing. Mixed-membership stochastic blockmodels. *Journal of Machine Learning Research*, 2008.

[2] S. Bengio and J. Mariéthoz. Learning the decision function for speaker verification. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2001.

[3] C.J.C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 1998.

[4] Aaron Clauset, Cristopher Moore, and M. E. J. Newman. Hierarchical structure and the prediction of missing links in networks. *Nature*, 2008.

[5] Andrew Gelman, Iwin Leenen, Iven Van Mechelen, Paul De Boeck, and Jeroen Poblome. Bridges between deterministic and probabilistic models for binary data. *Statistical Methodology*, 2010.

[6] L. Getoor and C.P. Diehl. Link mining: a survey. *ACM SIGKDD Explorations Newsletter*, 2005.

[7] M.S. Handcock, A.E. Raftery, and J.M. Tantrum. Model-based clustering for social networks. *Journal of the Royal Statistical Society: Series A*, 2007.

[8] Peter D Hoff, Adrian E Raftery, and Mark S Handcock. Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 2002.

[9] T. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. *Advances in Neural Information Processing Systems*, 1998.

[10] David Liben-Nowell and Jon Kleinberg. The link prediction problem for social networks. In *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management*, 2003.

[11] A.Y. Ng and M.I. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in Neural Information Processing Systems*, 2002.

[12] Adrian E. Raftery, Xiaoyue Niu, Peter Hoff, and Ka Yee Yeung. Fast inference for the latent space network model using a case-control approximate likelihood. Technical report, Center for Statistics and the Social Sciences, University of Washington, Seattle, USA, 2010.

[13] Michael Salter-Townshend and Thomas Brendan Murphy. Variational bayesian inference for the latent position cluster model. In *Workshop on Analyzing Networks and Learning with Graphs*. Neural Information Processing Systems, 2009.

[14] V.N. Vapnik. *Statistical learning theory*. Wiley New York, 1998.

[15] Zhou, Tao, Lü, Linyuan, and Zhang, Yi-Cheng. Predicting missing links via local information. *Eur. Phys. J. B*, 2009.