Side Information Aware Bayesian Affinity Estimation

Aayush Sharma Joydeep Ghosh

{asharma/ghosh}@ece.utexas.edu

IDEAL-2010-TR01*

Intelligent Data Exploration and Analysis Laboratory (IDEAL)

(Web: http://www.ideal.ece.utexas.edu/)

Department of Electrical and Computer Engineering
The University of Texas at Austin
Austin, Texas 78712
U.S.A.

August 4, 2010

Abstract

The estimation of affinity relationships between sets of entities can be aided by a variety of available sources of auxiliary information about these entities. Estimating the unknown affinity values given a few known values forms an important class of problems in data mining. This paper introduces a class of Bayesian mixture models - Side Information Aware Bayesian Affinity Estimation (SABAE), that efficiently exploits all the available sources of information within a common framework to predict affinity relationships. In particular, the models relate multiple information sources such as available past affinities, independent entity attributes (covariates), and/or a neighborhood structure over entities (e.g. a social network), to accurately predict missing affinities between each entity pair.

Utilizing side information allows seamless handling of both *warm-start* and *cold-start* within a single framework - affinities for previously unseen entities can be estimated using the auxiliary information associated with these 'new' entities. We further embed a factorized representation of affinities in SABAE to leverage the predictive power of matrix factorization based approaches. The resulting 'factorization aware Bayesian affinity estimation' helps to achieve superior predictive capability. A Bayesian approach further allows us to infer the missing side information for the entities conditioned on the available affinities along with other auxiliary information. In particular, we show how missing entity attributes can be estimated within the SABAE framework. The estimated attributes can then be leveraged for future affinity predictions.

Exploiting multiple sources of information entails the well-known feature selection problem. In this paper, we extend the SABAE framework for learning sparse homogeneous decompositions of the input affinity space that allows efficient feature selection from multiple available sources. Further, we provide efficient generative models for model selection among choices at varying resolutions. This joint feature and model selection framework results in more interpretable and actionable models.

An important property of several datasets describing affinity relationships is that the available affinities are often recorded in a self-selecting manner. We incorporate this self-selecting property into SABAE by explicitly modeling the probability of observing an affinity between pairs of entities. This *data observability model* helps in unbiased parameter estimation, thereby improving the prediction accuracy of the missing affinities. Yet another property of these datasets is their dynamic nature characterized by constant arrival and/or departure of entities, continuously evolving preferences, tastes and hence affinity relationships. We incorporate this dynamic nature of the data into a Bayesian affinity estimation framework with temporal dynamics resulting in better suited models for each individual time stamp.

Moreover in some applications, the learnt affinities are often used to generate a ranked preference list of one set of entities for another entity set. We further enhance the SABAE framework for a supervised ranking task that allows efficient learning of such rankings. Finally, we show how the models within the SABAE framework can be used in semi-supervised co-clustering settings, including an efficient modeling of the traditional must-link/cannot-link based constrained co-clustering. Efficient inferencing and learning algorithms for the SABAE framework based on variational mean field approximations are provided that allow scaling to large real-life datasets. Extensive experiments on simulated and real datasets illustrate the efficacy of the proposed models.

1 Introduction

Many datasets today contain affinity relationship information between two or more sets of entities. Estimating the unknown affinity values given a few known values forms an important class of problems in data mining. For example, in recent years, recommender systems have proved to be very successful at identifying affinities between users and items. Identifying personalized content of interest can greatly enrich the user experience and help institutions offering the recommendations to effectively target different kinds of users by predicting the propensity of users towards a set of items. Marketing data lends itself perfectly for an affinity estimation problem wherein effective marketing strategies can be formulated based on the predicted affinities. Additionally, there are useful applications in estimating affinities as clickthrough rates for online ads associated with users, search queries, or web pages. A common footing ground for all these applications is that the data arising in such domains generally consists of two sets of entities interacting with each other. Such data is known as dyadic data [1] and the goal is to predict the affinities between pairs of entities from the two sets. For example, in a user-movie recommendation engine, the interacting entities are sets of users and movies. The aim then, is to find the propensity rating for a user-movie pair.

Many current approaches for affinity estimation have concentrated only on a small number of known affinities to infer the missing ones ([2], [3], [4]). However, there are often available, many auxiliary sources of information associated with the entities that can aid the estimation of affinities between them. The most common source of additional information is the set of descriptive attributes associated with each entity. For example, in a movie recommendation engine, the attributes associated with a user might consist of demographic information such as age, gender, geo-location etc. that are often collected as profile information at the time of user registration. Similarly, movie attributes consist of readily available features such as genre, release date, running time, MPAA rating etc. The attributes associated with entities can have a strong bearing on the affinity relationships. For example, it may be common for adolescent males to enjoy movies about comic book characters. In this case, it could be very helpful to have the age and gender information of the user when attempting to predict the affinity of that user for such a movie. Another important source of auxiliary information about entities is a neighborhood structure over them. For example, users in a recommender system setting can also be a part of a social network represented by a user-user directed graph. The linkage structure can have an impact on a user's affinities, since preferences are often influenced by preferences of one's friends. Thus, one needs to efficiently account for these sources of information for accurate affinity estimation. In this paper, we introduce a class of Bayesian mixture models - Side Information Aware Bayesian Affinity Estimation (SABAE), that efficiently relates multiple sources of information such as available past affinities, entity attributes and/or neighborhood structure over entities, to accurately predict missing affinities between each entity pair.

Another problem associated with the methods relying only on past affinities is their inability to intelligently cater to affinity estimation for new entities with no prior history of affinities. This is referred to as a *cold-start* problem. The best one can do with these methods is to utilize a global average model, which however, fails to capture subtle correlations that might exist between a few existing and the new entities. Accurate prediction of affinities for new entities is very crucial for many applications. In the recommender system example, predicting affinities for a new product before its launch could help companies to use more targeted marketing techniques, and could help users recently introduced to the system to quickly find products that they will find useful. By efficiently utilizing the available side information, SABAE allows accurate prediction of affinities for new entities. Specifically, in the absence of past affinities, the available auxiliary information can be used to predict the missing affinities for these new entities.

Even though, a majority of the factorization based approaches fail to handle the cold-start problem, they are fairly effective in predicting affinities for the existing entities. This is evident in the plethora of methods that have been proposed in the context of the movie recommendation systems, particularly for the Netflix challenge [6]. However, generally it is hard to assign definitive interpretations to the learnt factorized representation of the affinities. The different factors do not necessarily have meaning in an external context, and it is difficult to use them to understand and explain the underlying causes for the observed affinity relationships. Clustering frameworks on the other hand yield more interpretable and actionable representations. In this paper we combine the best of both worlds - a factorized representation is embedded within the mixture model based SABAE framework to leverage the predictive power of the factorized representation and at the same time, retaining the interpretability of a clustering based mixture model.

Existing approaches to affinity estimation assume that the missing affinities are missing at random (MAR) [7]. However, this assumption is generally incorrect as is evident from a large number of affinity expressing datasets that are highly sparse with highly non-random distribution of observed affinities; a small fraction of entities account for a large fraction of the available data while the remaining observations are sparsely distributed among the remaining entities [8]. In a typical rating system, a user often chooses to rate a particular item if there is a strong positive or negative affinity for that item. This behavior suggests that that the observation of an affinity is dependent on the actual value of the affinity itself. Ignoring this dependence can result in biased models thereby, significantly impairing a model's affinity predictions. To overcome this problem, we propose a novel Bayesian framework that explicitly models the probability of observing an affinity, thereby getting away with the MAR assumption. The resulting observation aware Bayesian affinity estimation framework significantly improves the prediction accuracy of the missing affinities.

A key aspect of many affinity expressing datasets is their inherent dynamic nature. The preferences of entities are often changing with time. Further, the arrival of new entities or the removal of the existing ones can also greatly influence the affinity relationships. An accurate modeling of this evolution of preferences can greatly improve the prediction accuracy of the missing affinities [6]. We extend the SABAE framework within a state-space model of the input space for modeling the dynamic behavior of the data. An efficient algorithm based on variational Kalman filtering [9] is used to update the parameters of the state-space at each time step. This allows an efficient up to date modeling of the affinity relationships reflecting their evolving nature at each time step.

In addition to estimating missing affinities, the proposed SABAE framework can be efficiently extended to solve many related problems as a side product. In particular, a Bayesian framework allows us to estimate the missing entity attributes which is useful in a noisy data gathering process wherein the attributes are either missing or noisy. The estimated attributes can then be leveraged for future affinity predictions for the associated entities. The model for incorporating the neighborhood structure is applied to a semi-supervised co-clustering setting, including the traditional must-link/cannot-link constrained co-clustering and matrix approximation. Further, the Bayesian methodology allows us to learn generative models on the input affinity space resulting in an efficient feature and model selection framework. Finally, the framework is further extended to a supervised ranking task for learning an ordering on the missing affinities. Such an ordering can be used to generate preference lists for the entities (useful in displaying query search results or making top-k recommendations to users in a recommendation system).

1.1 Contributions

The key contributions of the paper can be summarized as follows:

1. A novel flexible Bayesian framework, Side Information Aware Bayesian Affinity Estimation (SABAE), that relates multiple sources of information - past affinities, entity attributes and/or neighborhood structure over entities, for an accurate affinity estimation in diverse data types. Utilizing the auxiliary information allows seamless handling of both warm-start and cold-start within a single framework.

2. An embedding framework for factorization and co-clustering based approaches that leverages superior predictive capability of a factorized representation together with the interpretability of a clustering framework. This results in highly accurate yet easily actionable and interpretable models for affinity estimation.

3. A novel observation aware affinity estimation framework, that efficiently models the absence or presence of affinities by moving away from the missing at random assumption and considering different reasons why an affinity may be missing.

4. A dynamic Bayesian affinity estimation framework that efficiently captures the evolution of affinity relationships between different entity pairs, resulting in accurate affinity predictions at each time stamp.

5. A generative model for learning sparse homogeneous decompositions of the input affinity space resulting in efficient feature selection from multiple available sources along with the suitable model selection among choices at varying resolutions.

6. A consistent supervised ranking framework that efficiently learns an ordering on the missing affinities to generate top-k preference lists. Such lists are widely used for making top-k recommendations to users in recommendation engines.

7. An extension of the SABAE framework to infer the missing entity attributes which can then be leveraged for future affinity predictions. Further, the paper shows how the models learnt within the SABAE framework can be used in semi-supervised co-clustering settings, including an efficient modeling of the traditional must-link/cannot-link based constrained co-clustering.

8. Efficient inferencing and learning algorithms based on a mean field approximation which ensure that all the frameworks are scalable to large-scale datasets by avoiding the sampling based MCMC methods.

The rest of the paper is organized as follows. We summarize some of the recent work for affinity estimation problems in 2. The basic SABAE framework is introduced in section 3 by explicit modeling of the entity attributes resulting in an attribute aware framework (AA-BAE). Neighborhood structure information is brought in section 4 to yield a neighborhood sensitive framework (NA-BAE). Factorized representation of the affinities is incorporated in section 5 resulting in a Factorization Aware Bayesian Affinity Estimation framework (FA-BAE). The data observability is modeled in section 6 to obtain observation aware (OA-BAE) framework while the temporal dynamics are modeled in section 7. Details on missing entity attributes estimation is included in section 8, sparse models for joint feature and model selection are learnt in 9 and rankings over the unknown affinities are learnt in section 10. We describe how NA-BAE model can be used for a semi-supervised co-clustering task in 11 and finally conclude in section 12.

2 Related Work

There is extensive literature in the data mining community for solving the problem of predicting unknown affinities. Recently, latent factor models have become popular and successful approaches for affinity estimation. Several authors have applied variants of these models to the Netflix problem ([3], [5], [6], [10]). For instance, Probabilistic Matrix Factorization (PMF) [3] is based on the idea that user-movie ratings can be represented as a product of user specific and movie specific latent factors. Different ways of regularizing the latent factors results in techniques with different properties and generalization capabilities. While these techniques scale well, are able to address the sparsity and imbalance of the Netflix data and improve accuracy, none of them consider any sort of auxiliary information. Moreover, none of these approaches can efficiently handle the cold-start problem. Recently, PMF was extended to a relational tensor factorization task [11], however the emphasis was still on the past relations.

Content based filtering is an alternative prediction technique, where the predictions are based on the entity attributes, e.g., annotations associated with books/blogs. Basilico and Hofmann [10] proposed a unified approach that integrates past affinities and entity attributes. Their main contribution is the design of a joint kernel over entity pairs that captures the similarity of past affinities as well as attribute information. While there has been some more work in hybrid algorithms([12], [1], [13]) and techniques to address cold-start problems [8], none of these approaches simultaneously handle auxiliary side information and data imbalance. Other efforts ([13], [1]) have concentrated on the entity attributes to simultaneously co-cluster the input space as well as learn multiple predictive models for predicting the unknown affinities, while a few others use the attributes information solely as moderating priors for the latent factor models ([8], [14]). While most of these approaches are able to handle cold-start problem by collapsing to models based on attributes, however the attributes only indirectly affect the affinities. This yields opaque models that are difficult to interpret.

Recently, attempts have been made to incorporate the social network data into the problem of predicting user-item ratings [15]. The authors incorporate user trust information into the PMF model and illustrate that it improves the accuracy of predicting ratings on the Epinions dataset [15]. However, obtaining or estimating trust information for all pairs of users is difficult and expensive when the only information available is largely just the social network's graph structure. Our neighborhood aware Bayesian affinity estimation framework only requires such a structure to obtain better prediction accuracies. Lu et al. [14] construct a nearest neighbor graph structure using the available user-movie attributes and then use the structure as a smoothing MRF prior for the PMF model. Though, this approach can be utilized to utilize other available neighborhood structures, it cannot simultaneously handle both attributes and the neighborhood structures. Several attempts have been made to include temporal dynamics into affinity estimation problems, especially in the context of the Netflix challenge. Koren [6] identifies different reasons for dynamic behavior of recommender systems and systematically incorporates the dynamics in a latent factor model. The model however, involves a large number of user-defined parameters which makes it very expensive to train using cross-validation. A linear state-space based model was proposed in [14] for dynamic user-item affinity prediction, wherein the user latent factors were smoothly evolved over time using a Gaussian process. However, dynamics over the items (such as item popularity etc.) were ignored.

It is generally assumed in approaches to affinity estimation that the missing data is distributed at random (often referred to as Missing at Random, or MAR [7]). However, in [16], the authors note that this assumption is often incorrect. If the data is not MAR, it has been shown in [17] that failing to model the distribution of missing data can significantly impair a model's affinity predictions. [16] models the missing data distribution using Conditional Probability Tables, based solely on the affinity values (which must be categorical). This approach is simplistic (and ad-hoc) and relies solely on the affinity values and fails to provide insights into the reasons for the missingness of an affinity. Other approaches, include the implicit feedback obtained as a result of the observation of an affinity ([18], [5]). We introduce an observation aware Bayesian affinity estimation framework that explicitly models the data missingness probabilities in an efficient and systematic manner, including an exploration of the reasons responsible for the data missingness.

Several Bayesian formulations have been proposed in the context of affinity estimation problems. Mixed Membership stochastic Blockmodels (MMBs) [19] is one such method that utilizes affinity values to group the two entity sets via a soft co-clustering. A weighted average of the pertinent co-cluster means is then used to estimate missing affinities. The model is shown to be quite efficient in scaling to large datasets, however it fails to utilize any available side information. Other efforts include fully Bayesian frameworks for PMF([20], [21]) with differing inference techniques - ranging from Variational approximations to sampling based MCMC methods. However, the stress again is only on utilizing the available affinities. Recently, Bayesian models based on topic models for document clustering [22] have been utilized for estimating affinities between users and News articles [23]. Two sided generalizations of topic models have also been utilized for co-clustering and matrix approximation problems ([24], [26]) without taking into account auxiliary sources of information. In this paper, we extend the topic models based approaches to systematically bring in different sources of information including past affinities, entity attributes, neighborhood structures, temporal dynamics and/or observability models. A taxonomy describing latent variables based approaches to affinity estimation can be obtained from Appendix E.

Notation. Before describing the SABAE framework, a quick word on the notation. We use capital script letters for sets, $\{\cdot\}$ denote a collection of variables for unnamed sets and $\dagger$ represents transpose of a matrix. Let $\mathcal{E}_1 = \{e_{1m}\}, [m]_1^M$ and $\mathcal{E}_2 = \{e_{2n}\}, [n]_1^N$ represent the sets of entities between which affinities need to be estimated. $\mathcal{Y} = \{y_{mn}\}$ is a set of $M \times N$ affinities between pairs of entities of the form $(e_{1m}, e_{2n}), e_{1m} \in \mathcal{E}_1$ and $e_{2n} \in \mathcal{E}_2$. The subset $\mathcal{Y}_{\text{obs}} \subseteq \mathcal{Y}$ is a set of observed affinities while $\mathcal{Y}_{\text{unobs}} = \mathcal{Y} \backslash \mathcal{Y}_{obs}$ denotes a set of missing affinities. A weight w_{mn} is associated with each affinity y_{mn} (affinity between a pair of entities e_{1m} and e_{2n}) such that $w_{mn} = 1$ if $y_{mn} \in \mathcal{Y}_{\text{obs}}$ and $w_{mn} = 0$ if $y_{mn} \in \mathcal{Y}_{\text{unobs}}$. The set of all $M \times N$ weights is denoted by $\mathcal{W}$. The set of entity attributes associated with $\mathcal{E}_1$ and $\mathcal{E}_2$ are respectively described by the sets $\mathcal{X}_1 = \{\boldsymbol{x}_{1m}\}$ and $\mathcal{X}_2 = \{\boldsymbol{x}_{2n}\}$. The notation $\boldsymbol{x}_{mn} = [\boldsymbol{x}_{1m}^{\dagger} \boldsymbol{x}_{2n}^{\dagger}]^{\dagger}$ is used to denote the attributes associated with the entity pair (e_{1m}, e_{2n}).

3 Attribute Aware Bayesian Affinity Estimation

This section introduces Side Information Aware Bayesian Affinity Estimation (SABAE), a generative framework for estimating affinities between two sets of entities. In this section, we only consider the available side information to be a set of attributes (covariates) associated with each entity. Additional sources of side information such as network structures over entities will be discussed in the next section. Figure 1 shows the graphical model for Attribute Aware Bayesian Affinity Estimation (AA-BAE) - a mixture model of KL clusters obtained as a cross-product of clustering the two sets of entities into K and L clusters respectively.

Each entity $e_{1m} \in \mathcal{E}_1$ is assigned to one of K clusters, by first sampling the mixing coefficients $\boldsymbol{\pi}_{1m}$ from a Dirichlet distribution $\text{Dir}(\boldsymbol{\alpha}_1)$. The cluster assignments $z_{1m} \in \{1, \ldots, K\}$ are then sampled

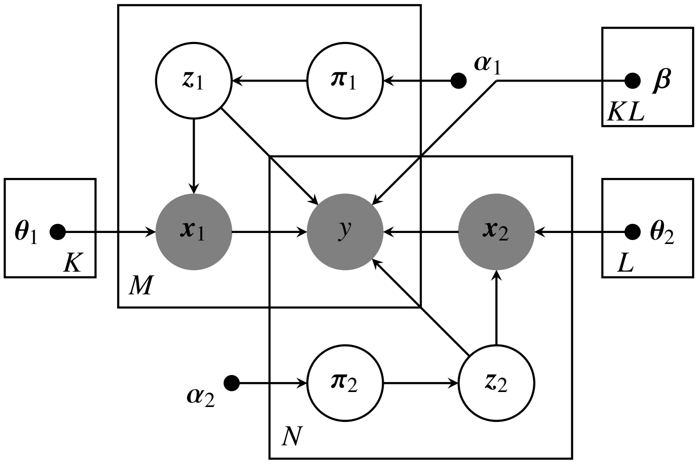

Figure 1: Graphical model for Attribute Aware Bayesian Affinity Estimation

from a discrete distribution Disc($\boldsymbol{\pi}_{1m}$) over the mixing coefficients. Similarly, the entities $e_{2n} \in \mathcal{E}_2$, are clustered into L clusters by first sampling the mixing coefficients $\boldsymbol{\pi}_{2n}$ from Dir($\boldsymbol{\alpha}_2$) followed by sampling cluster assignments $z_{2n} \in \{1, \ldots, L\}$ from a discrete distribution Disc($\boldsymbol{\pi}_{2n}$). We denote the set of mixing coefficients for the entities in $\mathcal{E}_1$ and $\mathcal{E}_2$ by $\boldsymbol{\pi}_1$ and $\boldsymbol{\pi}_2$ respectively. Similarly, $\mathcal{Z}_1$ and $\mathcal{Z}_2$ are respectively the sets of cluster assignments for the two entity sets.

The attributes $\boldsymbol{x}_{1m}$ associated with the entity e_{1m} are drawn from one of K possible exponential family distributions of the form $p_{\psi_1}(\boldsymbol{x}_{1m}|\boldsymbol{\theta}_{1z_{1m}})$[1], such that the parameter $\boldsymbol{\theta}_{1z_{1m}}$ of the family, is chosen according the entity cluster assignment z_{1m}. Likewise, attributes $\boldsymbol{x}_{2n}$ for an entity e_{2n} are generated from one of L possible exponential family distributions $p_{\psi_2}(\boldsymbol{x}_{2n}|\boldsymbol{\theta}_{2z_{2n}})$. The cluster assignments z_{1m} and z_{2n} over the two entities together determine a co-cluster (z_{1m}, z_{2n}), which then selects an exponential family distribution, $p_{\psi_Y}(y_{mn}|\boldsymbol{\beta}^{\dagger}_{z_{1m}z_{2n}}\boldsymbol{x}_{mn})$ (out of KL such distributions), to generate the affinity y_{mn} associated with the entity pair (e_{1m}, e_{2n}). The parameters $\boldsymbol{\beta}_{z_{1m}z_{2n}}$ of the distribution are specific to the co-cluster (z_{1m}, z_{2n}). In summary, the generative process for the attributes and the affinities between each pair of entities is as follows (Figure 1):

1. For each entity $e_{1m} \in \mathcal{E}_1$
   (a) Sample mixing coefficients: $\boldsymbol{\pi}_{1m} \sim \text{Dir}(\boldsymbol{\alpha}_1)$
   (b) Sample cluster assignment: $z_{1m} \sim \text{Disc}(\boldsymbol{\pi}_{1m})$
   (c) Sample entity attributes: $\boldsymbol{x}_{1m} \sim p_{\psi_1}(\boldsymbol{x}_{1m}|\boldsymbol{\theta}_{1z_{1m}})$
2. For each entity $e_{2n} \in \mathcal{E}_2$
   (a) Sample mixing coefficients: $\boldsymbol{\pi}_{2n} \sim \text{Dir}(\boldsymbol{\alpha}_2)$
   (b) Sample cluster assignment: $z_{2n} \sim \text{Disc}(\boldsymbol{\pi}_{2n})$
   (c) Sample entity attributes: $\boldsymbol{x}_{2n} \sim p_{\psi_2}(\boldsymbol{x}_{2n}|\boldsymbol{\theta}_{2z_{2n}})$
3. For each pair of entities (e_{1m}, e_{2n}) such that $e_{1m} \in \mathcal{E}_1, e_{2n} \in \mathcal{E}_2$
   (a) Sample affinity: $y_{mn} \sim p_{\psi_Y}(y_{mn}|\boldsymbol{\beta}^{\dagger}_{z_{1m}z_{2n}}\boldsymbol{x}_{mn})$

[1] We use the canonical form of exponential family distributions: $p_\psi(x|\theta) = p_0(x)\exp(\langle x, \theta\rangle - \psi(\theta))$

Note that within each co-cluster (k,l), $[k]_1^K$, $[l]_1^L$, the affinities are modeled via a *generalized linear model* [27] conditioned over the entity attributes. This property, along with the use of exponential family of distributions for modeling the attributes, provides a great flexibility in modeling diverse data types within a single framework. The overall joint distribution over all observable and latent variables is given by

$$p(\mathcal{Y},\mathcal{X}_1,\mathcal{X}_2,\mathcal{Z}_1,\mathcal{Z}_2,\boldsymbol{\pi}_1,\boldsymbol{\pi}_2|\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2,\boldsymbol{\Theta}_1,\boldsymbol{\Theta}_2,\boldsymbol{\beta}) = \tag{1}$$

$$\left(\prod_m p(\boldsymbol{\pi}_{1m}|\boldsymbol{\alpha}_1)p(z_{1m}|\boldsymbol{\pi}_{1m})p_{\psi_1}(\boldsymbol{x}_{1m}|\boldsymbol{\theta}_{1z_{1m}})\right)\left(\prod_n p(\boldsymbol{\pi}_{2n}|\boldsymbol{\alpha}_2)p(z_{2n}|\boldsymbol{\pi}_{2n})p_{\psi_2}(\boldsymbol{x}_{2n}|\boldsymbol{\theta}_{2z_{2n}})\right)\left(\prod_{m,n} p_{\psi_Y}(y_{mn}|\boldsymbol{\beta}^{\dagger}_{z_{1m}z_{2n}}\boldsymbol{x}_{mn})\right)$$

Marginalizing out the latent variables, the probability of observing the known affinities and the attributes is:

$$p(\mathcal{Y}_{\text{obs}},\mathcal{X}_1,\mathcal{X}_2|\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2,\boldsymbol{\Theta}_1,\boldsymbol{\Theta}_2,\boldsymbol{\beta}) = \int_{\mathcal{Y}_{\text{unobs}}}\int_{\boldsymbol{\pi}_1}\int_{\boldsymbol{\pi}_2}\left(\prod_m p(\boldsymbol{\pi}_{1m}|\boldsymbol{\alpha}_1)\right)\left(\prod_n p(\boldsymbol{\pi}_{2n}|\boldsymbol{\alpha}_2)\right) \tag{2}$$

$$\sum_{\mathcal{Z}_1}\sum_{\mathcal{Z}_2}\left(\prod_m p(z_{1m}|\boldsymbol{\pi}_{1m})p_{\psi_1}(\boldsymbol{x}_{1m}|\boldsymbol{\theta}_{1z_{1m}})\right)\left(\prod_n p(z_{2n}|\boldsymbol{\pi}_{2n})p_{\psi_2}(\boldsymbol{x}_{2n}|\boldsymbol{\theta}_{2z_{2n}})\right)\left(\prod_{m,n} p_{\psi_Y}(y_{mn}|\boldsymbol{\beta}^{\dagger}_{z_{1m}z_{2n}}\boldsymbol{x}_{mn})\right)d\mathcal{Y}_{\text{unobs}}d\boldsymbol{\pi}_1 d\boldsymbol{\pi}_2$$

It is easy to see that AA-BAE extends the Bayesian co-clustering (BCC) [24] to a prediction framework by explicitly modeling the attributes associated with the entities. A crucial departure from BCC (and from most other mixture models) is that cluster assignment latent variables z_{1m}, z_{2n} are sampled only once for entities e_{1m} and e_{2n} respectively. The assignments then combine to generate the set of affinities $y_{m\cdot}$ and $y_{\cdot n}$ thereby inducing coupling between these affinities. In contrast, in BCC cluster assignments are sampled for every entry y_{mn}.

3.1 Inference and Learning

The model parameters $(\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2,\boldsymbol{\Theta}_1,\boldsymbol{\Theta}_2,\boldsymbol{\beta})$ can in theory, be learnt by maximizing the observed log-likelihood in equation (3) using the expectation maximization (EM) family of algorithms [28]. However, computation of $\log p(\mathcal{Y}_{\text{obs}},\mathcal{X}_1,\mathcal{X}_2|\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2,\boldsymbol{\Theta}_1,\boldsymbol{\Theta}_2,\boldsymbol{\beta})$ is intractable for AA-BAE, rendering direct application of EM infeasible. To overcome this problem, we propose a variational mean field approximation [29] to the true posterior distribution of the latent variables. This allows us to construct tractable lower bounds on the observed likelihood, which can be efficiently maximized with respect to the model parameters.

3.1.1 Inference using Naïve Mean Field Approximation

To get a tractable lower bound, we approximate the true posterior distribution over the latent variables by a fully factorized distribution of the following form:

$$q(\mathcal{Y}_{\text{unobs}},\mathcal{Z}_1,\mathcal{Z}_2,\boldsymbol{\pi}_1,\boldsymbol{\pi}_2) = \left(\prod_{\substack{m,n\\ y_{mn}\in\mathcal{Y}_{\text{unobs}}}} q(y_{mn})\right)\left(\prod_m q(\boldsymbol{\pi}_{1m})q(z_{1m})\right)\left(\prod_n q(\boldsymbol{\pi}_{2n})q(z_{2n})\right) \tag{3}$$

Applying Jensen's inequality [30], the following lower bound then exists for the observed log-likelihood:

$$\log p(\mathcal{Y}_{\text{obs}},\mathcal{X}_1,\mathcal{X}_2|\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2,\boldsymbol{\Theta}_1,\boldsymbol{\Theta}_2,\boldsymbol{\beta}) \geq H[q]+\mathbb{E}_q[\log p(\mathcal{Y}_{\text{obs}},\mathcal{X}_1,\mathcal{X}_2,\mathcal{Z}_1,\mathcal{Z}_2,\boldsymbol{\pi}_1,\boldsymbol{\pi}_2|\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2,\boldsymbol{\Theta}_1,\boldsymbol{\Theta}_2,\boldsymbol{\beta})] \tag{4}$$

where $H[q]$ is the entropy of the variational distribution q while $\mathbb{E}_q[\cdot]$ is the expectation with respect to q. Let Q be a set of all distributions having a fully factorized form (3). Among all variational distributions $q \in Q$, we then seek a distribution that provides the tightest lower bound on the observed log-likelihood. The optimal distribution corresponding the tightest lower bound is then given by:

$$q^* = \arg\max_{q \in Q} H[q] + \mathbb{E}_q[\log p(\mathcal{Y}_{\text{obs}}, \mathcal{X}_1, \mathcal{X}_2, \mathcal{Z}_1, \mathcal{Z}_2, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2 | \boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2, \boldsymbol{\beta})] \tag{5}$$

Let i be an index variable corresponding to individual factors of the variational distribution in (3) such that $q = \prod_i q_i$. The optimal solution for the factor q_i^* then assumes a Gibbs' distribution of the following form([31], Chapter 10):

$$q_i^* = \frac{1}{\Upsilon_i} \exp\left(\mathbb{E}_{q|q_i}[\log p(\mathcal{Y}_{\text{obs}}, \mathcal{X}_1, \mathcal{X}_2, \mathcal{Z}_1, \mathcal{Z}_2, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2 | \boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2, \boldsymbol{\beta})]\right) \tag{6}$$

$\mathbb{E}_{q|q_i}[\cdot]$ is the conditional expection with respect to q conditioned on the factor q_i and Υ_i is the normalization constant.

Using (6), the optimal variational distribution over the latent variables then assumes the following form:

$$q^*(\mathcal{Y}_{\text{unobs}}, \mathcal{Z}_1, \mathcal{Z}_2, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2) = \tag{7}$$

$$\left(\prod_{\substack{m,n \\ y_{mn} \in \mathcal{Y}_{\text{unobs}}}} q^*_{\psi_y}(y_{mn}|\phi_{mn})\right)\left(\prod_m q^*(\boldsymbol{\pi}_{1m}|\boldsymbol{\gamma}_{1m}) q^*(z_{1m}|\boldsymbol{r}_{1m})\right)\left(\prod_n q^*(\boldsymbol{\pi}_{2n}|\boldsymbol{\gamma}_{2n}) q^*(z_{2n}|\boldsymbol{r}_{2n})\right)$$

where $q_{\psi_y}(y_{mn}|\phi_{mn})$ is an exponential family distribution of the same form as the one assumed for the affinities and with natural parameter ϕ_{mn}, $q(\boldsymbol{\pi}_{1m}|\boldsymbol{\gamma}_{1m})$ and $q(\boldsymbol{\pi}_{2n}|\boldsymbol{\gamma}_{2n})$ are K and L dimensional Dirichlet distributions with parameters $\boldsymbol{\gamma}_{1m}, \boldsymbol{\gamma}_{2n}$ respectively. Variational distributions over cluster assignments $q(z_{1m}|\boldsymbol{r}_{1m})$ and $q(z_{2n}|\boldsymbol{r}_{2n})$ follow discrete distributions over K and L clusters with parameters $\boldsymbol{r}_{1m}, \boldsymbol{r}_{2n}$ respectively. $(\boldsymbol{\phi}, \boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2, \boldsymbol{r}_1, \boldsymbol{r}_2)$ are collectively known as the *variational parameters*. The variational parameters satisfying (6) are as follows:

$$\phi_{mn} = \sum_{k=1}^{K} \sum_{l=1}^{L} r_{1mk} r_{2nl} \left(\boldsymbol{\beta}_{kl}^{\dagger} \boldsymbol{x}_{mn}\right) \tag{8}$$

$$\gamma_{1mk} = r_{1mk} + \alpha_{1k} \tag{9}$$

$$\gamma_{2nl} = r_{2nl} + \alpha_{2l} \tag{10}$$

$$r_{1mk} \propto \exp\left(\log p_{\psi_1}(\boldsymbol{x}_{1m}|\boldsymbol{\theta}_{1k}) + \Psi(\gamma_{1mk}) + \sum_{n=1}^{N} \sum_{l=1}^{L} r_{2nl} \left(w_{mn} \log p_{\psi_y}(y_{mn}|\boldsymbol{\beta}_{kl}^{\dagger} \boldsymbol{x}_{mn}) + (1 - w_{mn}) \mathbb{E}_q \left[\log p_{\psi_y}(y_{mn}|\boldsymbol{\beta}_{kl}^{\dagger} \boldsymbol{x}_{mn})\right]\right)\right) \tag{11}$$

$$r_{2nl} \propto \exp\left(\log p_{\psi_2}(\boldsymbol{x}_{2n}|\boldsymbol{\theta}_{2l}) + \Psi(\gamma_{2nl}) + \sum_{m=1}^{M} \sum_{k=1}^{K} r_{1mk} \left(w_{mn} \log p_{\psi_y}(y_{mn}|\boldsymbol{\beta}_{kl}^{\dagger} \boldsymbol{x}_{mn}) + (1 - w_{mn}) \mathbb{E}_q \left[\log p_{\psi_y}(y_{mn}|\boldsymbol{\beta}_{kl}^{\dagger} \boldsymbol{x}_{mn})\right]\right)\right) \tag{12}$$

The expectation $\mathbb{E}_q[\log p_{\psi_y}(y_{mn}|\boldsymbol{\beta}_{kl}^{\dagger} \boldsymbol{x}_{mn})]$ is computed for unobserved affinities with respect to the varational exponential family distribution $q_{\psi_y}(y_{mn}|\phi_{mn})$. r_{1mk} can be interpreted as the *responsibility* of the k^{th} cluster for entity e_{1m}. Similarly, r_{2nl} represents the responsibility of the l^{th} cluster for the entity e_{2n}. γ_{1mk} is the k^{th} component of the Dirichlet distribution parameters $\boldsymbol{\gamma}_{1m}$ while γ_{2nl} is the l^{th}

Algorithm 1 Learn AA-BAE

```
Input: $\mathcal{Y}_{\text{obs}}, \mathcal{X}_1, \mathcal{X}_2, K, L$
Output: $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2, \boldsymbol{\beta}$
        $[m]_1^M, [n]_1^N, [k]_1^K, [l]_1^L$

  Step 0: Initialize $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2, \boldsymbol{\beta}$
  Until Convergence
      Step 1: E-Step
          Step 1a: Initialize $r_{1mk}, r_{2nl}$
          Until Convergence
              Step 1b: Update $\phi_{mn}$ using equation (8)
              Step 1c: Update $\gamma_{1mk}$ using equation (9)
              Step 1d: Update $\gamma_{2nl}$ using equation (10)
              Step 1e: Update $r_{1mk}$ using equation (11)
              Step 1f: Update $r_{2nl}$ using equation (12)
      Step 2: M-Step
          Step 2a: Update $\boldsymbol{\theta}_{1k}$ using equation (13)
          Step 2b: Update $\boldsymbol{\theta}_{2l}$ using equation (14)
          Step 2c: Update $\boldsymbol{\beta}_{kl}$ using equation (15)
          Step 2d: Update $\boldsymbol{\alpha}_1$ using equation (16)
          Step 2e: Update $\boldsymbol{\alpha}_2$ using equation (17)
```

component of the Dirichlet distribution parameters $\boldsymbol{\gamma}_{2n}$, and $\Psi(\cdot)$ is the digamma function. The set of coupled update equations (8) through (12) for the variational parameters are collectively known as *mean field equations* and can be satisfied iteratively. Often, to avoid the local minima problem, deterministic annealing [32] with geometric cooling schedule is used to update the cluster assignment parameters $\{r_{1m}, r_{2n}\}, [m]_1^M, [n]_1^N$.

3.1.2 Parameter Estimation

The optimized lower bound obtained from the inference step can then be maximized with respect to the free model parameters. Taking partial derivatives of the bound with respect to the model parameters $(\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2, \boldsymbol{\beta})$ and setting them to zero, the following updates for the parameters can be obtained:

$$\boldsymbol{\theta}_{1k} = \nabla\psi_1^{-1}\left(\frac{\sum_{m=1}^M r_{1mk}\boldsymbol{x}_{1m}}{\sum_{m=1}^M r_{1mk}}\right) \tag{13}$$

$$\boldsymbol{\theta}_{2l} = \nabla\psi_2^{-1}\left(\frac{\sum_{n=1}^N r_{2nl}\boldsymbol{x}_{2n}}{\sum_{n=1}^N r_{2nl}}\right) \tag{14}$$

$$\boldsymbol{\beta}_{kl} = \underset{\beta\in\mathbb{R}^D}{\arg\max} \sum_{m=1}^M \sum_{n=1}^N r_{1mk} r_{2nl}\left[\left\langle\left(w_{mn}y_{mn} + (1-w_{mn})\nabla\psi_{\mathcal{Y}}(\phi_{mn})\right), \boldsymbol{\beta}^\dagger \boldsymbol{x}_{mn}\right\rangle - \psi_{\mathcal{Y}}\left(\boldsymbol{\beta}^\dagger \boldsymbol{x}_{mn}\right)\right] \tag{15}$$

$$\boldsymbol{\alpha}_1 = \underset{\boldsymbol{\alpha}_1\in\mathbb{R}_{++}^K}{\arg\max} \sum_{m=1}^M \left(\log\frac{\Gamma(\sum_{k=1}^K \alpha_{1k})}{\prod_{k=1}^K \Gamma(\alpha_{1k})} + \sum_{k=1}^K(\alpha_{1k} + r_{1mk} - 1)\left(\Psi(\gamma_{1mk}) - \Psi\left(\sum_{k'=1}^K \gamma_{1mk'}\right)\right)\right) \tag{16}$$

$$\boldsymbol{\alpha}_2 = \underset{\boldsymbol{\alpha}_2\in\mathbb{R}_{++}^L}{\arg\max} \sum_{n=1}^N \left(\log\frac{\Gamma(\sum_{l=1}^L \alpha_{2l})}{\prod_{l=1}^L \Gamma(\alpha_{2l})} + \sum_{l=1}^L(\alpha_{2l} + r_{2nl} - 1)\left(\Psi(\gamma_{2nl}) - \Psi\left(\sum_{l'=1}^L \gamma_{2nl'}\right)\right)\right) \tag{17}$$

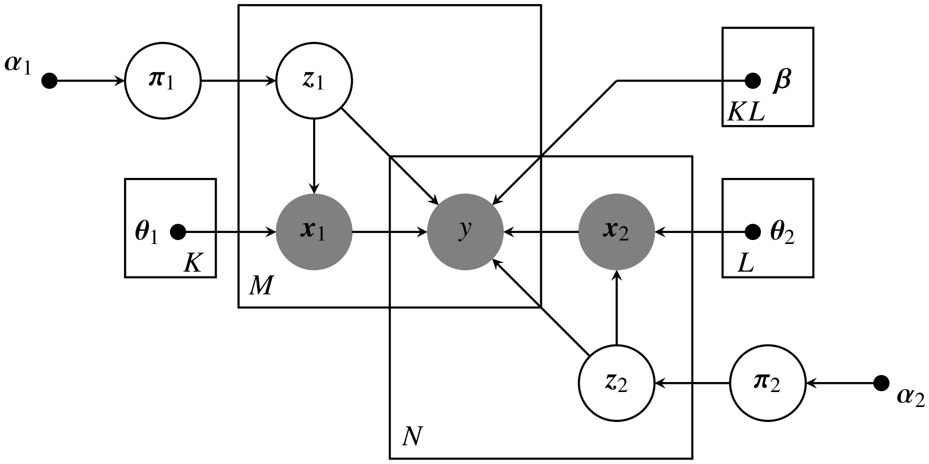

Figure 2: Graphical model for Latent Dirichlet Attribute Aware Bayesian Affinity Estimation

The updates for the natural parameters $\boldsymbol{\theta}_{1k}, \boldsymbol{\theta}_{2l}$ follow from the conjugacy[2] of the mean parameter and the natural parameter for an exponential family distribution [4]. The update for $\boldsymbol{\beta}_{kl}$ is a weighted GLM regression [27] over the attributes $\{\boldsymbol{x}_{mn}\}$ for the dependent variables set to the actual affinity y_{mn} if $y_{mn} \in \mathcal{Y}_{\text{obs}}$, i.e. $w_{mn} = 1$, while if the affinity is missing i.e. $y_{mn} \in \mathcal{Y}_{\text{unobs}}$ or $w_{mn} = 0$, the value is replaced by its expected value $\nabla\psi_{\mathcal{Y}}(\phi_{mn})$ under the variational exponential family distribution $q_{\psi_{\mathcal{Y}}}(y_{mn}|\phi_{mn})$. The weights in the weighted regression are the co-cluster responsibilities given by $r_{1mk}r_{2nl}$. Any off-the-shelf regression software can be used to efficiently update $\boldsymbol{\beta}_{kl}$ (e.g., `glmfit` function in matlab). Further, any form of convex regularization such as ℓ-1, ℓ-2 can be used to prevent overfitting in the $\boldsymbol{\beta}_{kl}$. Finally, we see that the parameters of the Dirichlet distribution $(\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2)$ can be efficiently learnt using the Newton-Raphson's method. Further, to constrain the parameters $\boldsymbol{\alpha}_1(\boldsymbol{\alpha}_2)$ to K (L) simplex, one can follow an adaptive line search strategy employed in [24].

Variational mean field approximation leads to an EM style algorithm wherein E-step consists of constructing a tight lower bound to the observed log-likelihood for fixed values of the model parameters. The optimized lower bound is then maximized with respect to the free model parameters in the subsequent M-step to get an improved estimate of the parameters. Starting with an initial guess of the model parameters, the algorithm iterates between the two steps till convergence. The resulting algorithm to learn the parameters of AA-BAE is given in algorithm 1.

3.2 Latent Dirichlet Attribute Aware Bayesian Affinity Estimation

In the generative process for AA-BAE model, the mixing coefficients $\boldsymbol{\pi}_{1m}(\boldsymbol{\pi}_{2n})$ are sampled once for every entity $e_{1m}(e_{2n})$ from the prior Dirichlet distributions $\text{Dir}(\boldsymbol{\alpha}_1)(\text{Dir}(\boldsymbol{\alpha}_2))$. Hence, conditioned of the parameters of the two Dirichlet distributions, the cluster assignment variables $z_{1m}(z_{2n})$ are sampled independently for every entity $e_{1m}(e_{2n})$. The generative process is thus, unable to capture the dependencies between different entities due to these independent cluster assignments.

To overcome this problem, we induce dependencies between the cluster assignments by sampling the mixing coefficients $\boldsymbol{\pi}_1(\boldsymbol{\pi}_2)$ only once for entity set $\mathcal{E}_1(\mathcal{E}_2)$. Hence, all the entities in a particular set share the same mixing coefficients, thereby inducing statistical dependency between them. Once

[2] For an exponential family distribution $p_\psi(x|\theta)$, the expected value follows: $\mathbb{E}[x] = \nabla\psi(\theta)$

the mixing coefficients are known, the cluster assignments are then sampled independently by discrete distributions over these mixing coefficients. It is easy to see that by sharing mixing coefficients across entities in a set, the model is an attribute sensitive two sided generalization of the Latent Dirichlet Allocation (LDA) [22] model. Hence, the generative process for ‘Latent Dirichlet Attribute Aware Bayesian Affinity Estimation (LD-AA-BAE)’ is then given by (figure 2):

1. Sample mixing coefficients: $\boldsymbol{\pi}_1 \sim \text{Dir}(\boldsymbol{\alpha}_1)$
2. Sample mixing coefficients: $\boldsymbol{\pi}_2 \sim \text{Dir}(\boldsymbol{\alpha}_2)$
3. For each entity $e_{1m} \in \mathcal{E}_1$
   (a) Sample cluster assignment: $z_{1m} \sim \text{Disc}(\boldsymbol{\pi}_1)$
   (b) Sample entity attributes: $\boldsymbol{x}_{1m} \sim p_{\psi_1}(\boldsymbol{x}_{1m}|\boldsymbol{\theta}_{1z_{1m}})$
4. For each entity $e_{2n} \in \mathcal{E}_2$
   (a) Sample cluster assignment: $z_{2n} \sim \text{Disc}(\boldsymbol{\pi}_2)$
   (b) Sample entity attributes: $\boldsymbol{x}_{2n} \sim p_{\psi_2}(\boldsymbol{x}_{2n}|\boldsymbol{\theta}_{2z_{2n}})$
5. For each pair of entities (e_{1m}, e_{2n}) such that $e_{1m} \in \mathcal{E}_1, e_{2n} \in \mathcal{E}_2$
   (a) Sample affinity: $y_{mn} \sim p_{\psi_{\mathcal{Y}}}(y_{mn}|\boldsymbol{\beta}^{\dagger}_{z_{1m}z_{2n}}\boldsymbol{x}_{mn})$

The overall joint distribution over all observable and latent variables is then given by:

$$p(\mathcal{Y}, \mathcal{X}_1, \mathcal{X}_2, \mathcal{Z}_1, \mathcal{Z}_2, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2|\alpha_1, \alpha_2, \boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2, \boldsymbol{\beta}) =$$

$$p(\boldsymbol{\pi}_1|\alpha_1)p(\boldsymbol{\pi}_2|\alpha_2)\left(\prod_m p(z_{1m}|\boldsymbol{\pi}_1)p_{\psi_1}(\boldsymbol{x}_{1m}|\boldsymbol{\theta}_{1z_{1m}})\right)\left(\prod_n p(z_{2n}|\boldsymbol{\pi}_2)p_{\psi_2}(\boldsymbol{x}_{2n}|\boldsymbol{\theta}_{2z_{2n}})\right)\left(\prod_{m,n} p_{\psi_{\mathcal{Y}}}(y_{mn}|\boldsymbol{\beta}^{\dagger}_{z_{1m}z_{2n}}\boldsymbol{x}_{mn})\right)$$

Marginalizing out the latent variables, the probability of observing the known affinities and the attributes is:

$$p(\mathcal{Y}_{\text{obs}}, \mathcal{X}_1, \mathcal{X}_2|\alpha_1, \alpha_2, \boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2, \boldsymbol{\beta}) = \int_{\mathcal{Y}_{\text{unobs}}}\int_{\boldsymbol{\pi}_1}\int_{\boldsymbol{\pi}_2}(p(\boldsymbol{\pi}_1|\alpha_1))\,(p(\boldsymbol{\pi}_2|\alpha_2))$$

$$\sum_{\mathcal{Z}_1}\sum_{\mathcal{Z}_2}\left(\prod_m p(z_{1m}|\boldsymbol{\pi}_1)p_{\psi_1}(\boldsymbol{x}_{1m}|\boldsymbol{\theta}_{1z_{1m}})\right)\left(\prod_n p(z_{2n}|\boldsymbol{\pi}_2)p_{\psi_2}(\boldsymbol{x}_{2n}|\boldsymbol{\theta}_{2z_{2n}})\right)\left(\prod_{m,n} p_{\psi_{\mathcal{Y}}}(y_{mn}|\boldsymbol{\beta}^{\dagger}_{z_{1m}z_{2n}}\boldsymbol{x}_{mn})\right)d\mathcal{Y}_{\text{unobs}}d\boldsymbol{\pi}_1 d\boldsymbol{\pi}_2$$

Note that even marginalization of only the mixing coefficients $\boldsymbol{\pi}_1$ and $\boldsymbol{\pi}_2$ induces dependencies between the clustering assignments $\mathcal{Z}_1$ and $\mathcal{Z}_2$.

3.2.1 Inference and Learning

As a result of the induced dependencies, direct maximization of the observed log-likelihood is intractable using an EM algorithm. Hence, we construct tractable lower bounds using a fully factorized mean field approximation to the true posterior distribution over the latent variables. Following analysis of section 3.1.1, the optimal factorized distribution over the latent variables $(\mathcal{Y}_{\text{unobs}}, \mathcal{Z}_1, \mathcal{Z}_2, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2)$ that corresponds to the tightest lower bound on the observed likelihood is then given by:

$$q^*(\mathcal{Y}_{\text{unobs}}, \mathcal{Z}_1, \mathcal{Z}_2, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2) = q^*(\boldsymbol{\pi}_1|\gamma_1)q^*(\boldsymbol{\pi}_{2|}\gamma_2)\left(\prod_{\substack{m,n \\ y_{mn}\in\mathcal{Y}_{\text{unobs}}}} q^*(y_{mn}|\phi_{mn})\right)\left(\prod_m q^*(z_{1m}|r_{1m})\right)\left(\prod_n q^*(z_{2n}|r_{2n})\right) \tag{18}$$

Note that, since the mixing coefficients are shared across entities from the same set, we only have two variational factors corresponding to the mixing coefficients $\boldsymbol{\pi}_1$ and $\boldsymbol{\pi}_2$. On the other hand, for AA-BAE there are $(M + N)$ variational factors for mixing coefficients, one for each entity in the two sets. The rest of the model however is similar to AA-BAE. As such, $q_{\psi_y}(y_{mn}|\phi_{mn})$ is an exponential family distribution with natural parameter ϕ_{mn}, $q(\boldsymbol{\pi}_1|\gamma_1)$ and $q(\boldsymbol{\pi}_2|\gamma_2)$ are K and L dimensional Dirichlet distributions with parameters γ_1 and γ_2 respectively while the cluster assignments z_{1m} and z_{2n} follow discrete distributions over K and L clusters with parameters r_{1m} and r_{2n} respectively. The variational parameters $(\gamma_1, \gamma_2, \phi_{mn}, r_{1m}, r_{2n})$ are then given by (see Appendix B for derivation):

$$\phi_{mn} = \sum_{k=1}^{K}\sum_{l=1}^{L} r_{1mk}r_{2nl}\left(\boldsymbol{\beta}_{kl}^{\dagger}\boldsymbol{x}_{mn}\right) \tag{19}$$

$$\gamma_{1k} = \sum_{m=1}^{M} r_{1mk} + \alpha_{1k} \tag{20}$$

$$\gamma_{2l} = \sum_{n=1}^{N} r_{2nl} + \alpha_{2l} \tag{21}$$

$$r_{1mk} \propto \exp\left(\log p_{\psi_1}(\boldsymbol{x}_{1m}|\boldsymbol{\theta}_{1k}) + \Psi(\gamma_{1k}) + \sum_{n=1}^{N}\sum_{l=1}^{L} r_{2nl}\left(w_{mn}\log p_{\psi_y}(y_{mn}|\boldsymbol{\beta}_{kl}^{\dagger}\boldsymbol{x}_{mn}) + (1-w_{mn})\mathbb{E}_q\left[\log p_{\psi_y}(y_{mn}|\boldsymbol{\beta}_{kl}^{\dagger}\boldsymbol{x}_{mn})\right]\right)\right) \tag{22}$$

$$r_{2nl} \propto \exp\left(\log p_{\psi_2}(\boldsymbol{x}_{2n}|\boldsymbol{\theta}_{2l}) + \Psi(\gamma_{2l}) + \sum_{m=1}^{M}\sum_{k=1}^{K} r_{1mk}\left(w_{mn}\log p_{\psi_y}(y_{mn}|\boldsymbol{\beta}_{kl}^{\dagger}\boldsymbol{x}_{mn}) + (1-w_{mn})\mathbb{E}_q\left[\log p_{\psi_y}(y_{mn}|\boldsymbol{\beta}_{kl}^{\dagger}\boldsymbol{x}_{mn})\right]\right)\right) \tag{23}$$

The optimal lower bound on the observed log-likelihood with respect to the variational distribution in (18) is then given by:

$$\log p(\mathcal{Y}_{\text{obs}}, \mathcal{X}_1, \mathcal{X}_2|\alpha_1, \alpha_2, \boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2, \boldsymbol{\beta}) \geq H[q^*]+\mathbb{E}_{q^*}[\log p(\mathcal{Y}_{\text{obs}}, \mathcal{X}_1, \mathcal{X}_2, \mathcal{Z}_1, \mathcal{Z}_2, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2|\alpha_1, \alpha_2, \boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2, \boldsymbol{\beta})]$$

This bound can be maximized with respect to the free model parameters to get their improved estimates. Taking partial derivatives of the bound with respect to the model parameters and setting them to zero, we obtain the following updates (see Appendix B for details):

$$\boldsymbol{\theta}_{1k} = \nabla\psi_1^{-1}\left(\frac{\sum_{m=1}^{M} r_{1mk}\boldsymbol{x}_{1m}}{\sum_{m=1}^{M} r_{1mk}}\right) \tag{24}$$

$$\boldsymbol{\theta}_{2l} = \nabla\psi_2^{-1}\left(\frac{\sum_{n=1}^{N} r_{2nl}\boldsymbol{x}_{2n}}{\sum_{n=1}^{N} r_{2nl}}\right) \tag{25}$$

Algorithm 2 Learn LD-AA-BAE

```
Input: $\mathcal{Y}_{\text{obs}}, \mathcal{X}_1, \mathcal{X}_2, K, L$
Output: $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2, \boldsymbol{\beta}$
        $[m]_1^M, [n]_1^N, [k]_1^K, [l]_1^L$

  Step 0: Initialize $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2, \boldsymbol{\beta}$
  Until Convergence
      Step 1: E-Step
          Step 1a: Initialize $r_{1mk}, r_{2nl}$
          Until Convergence
              Step 1b: Update $\phi_{mn}$ using equation (19)
              Step 1c: Update $\gamma_{1k}$ using equation (20)
              Step 1d: Update $\gamma_{2l}$ using equation (21)
              Step 1e: Update $r_{1mk}$ using equation (22)
              Step 1f: Update $r_{2nl}$ using equation (23)
      Step 2: M-Step
          Step 2a: Update $\boldsymbol{\theta}_{1k}$ using equation (24)
          Step 2b: Update $\boldsymbol{\theta}_{2l}$ using equation (25)
          Step 2c: Update $\boldsymbol{\beta}_{kl}$ using equation (26)
          Step 2d: Update $\boldsymbol{\alpha}_1$ using equation (27)
          Step 2e: Update $\boldsymbol{\alpha}_2$ using equation (28)
```

$$\boldsymbol{\beta}_{kl} = \underset{\beta\in\mathbb{R}^D}{\arg\max} \sum_{m=1}^{M} \sum_{n=1}^{N} r_{1mk} r_{2nl} \left[\left\langle (w_{mn} y_{mn} + (1-w_{mn})\nabla\psi_{\mathcal{Y}}(\phi_{mn})), \boldsymbol{\beta}^{\dagger}\boldsymbol{x}_{mn}\right\rangle - \psi_{\mathcal{Y}}\left(\boldsymbol{\beta}^{\dagger}\boldsymbol{x}_{mn}\right)\right] \tag{26}$$

$$\boldsymbol{\alpha}_1 = \underset{\boldsymbol{\alpha}_1\in\mathbb{R}^K_{++}}{\arg\max} \left(\log \frac{\Gamma(\sum_{k=1}^K \alpha_{1k})}{\prod_{k=1}^K \Gamma(\alpha_{1k})} + \sum_{k=1}^{K}\left(\alpha_{1k} + \sum_{m=1}^{M} r_{1mk} - 1\right)\left(\Psi(\gamma_{1k}) - \Psi\left(\sum_{k'=1}^{K}\gamma_{1k'}\right)\right)\right) \tag{27}$$

$$\boldsymbol{\alpha}_2 = \underset{\boldsymbol{\alpha}_2\in\mathbb{R}^L_{++}}{\arg\max} \left(\log \frac{\Gamma(\sum_{l=1}^L \alpha_{2l})}{\prod_{l=1}^L \Gamma(\alpha_{2l})} + \sum_{l=1}^{L}\left(\alpha_{2l} + \sum_{n=1}^{N} r_{2nl} - 1\right)\left(\Psi(\gamma_{2l}) - \Psi\left(\sum_{l'=1}^{L}\gamma_{2l'}\right)\right)\right) \tag{28}$$

Note that the form of updates for the parameters $(\boldsymbol{\theta}_{1k}, \boldsymbol{\theta}_{2l}, \boldsymbol{\beta}_{kl})$ is similar to ones obtained for AA-BAE. The updates for the parameters of the Dirichlet distributions $\boldsymbol{\alpha}_1$ and $\boldsymbol{\alpha}_2$, can be efficiently performed using the Newton-Raphson's method. An EM algorithm for learning the model parameters of LD-AA-BAE is given in algorithm 2.

4 Neighborhood Aware Bayesian Affinity Estimation

An important source of auxiliary information for affinity expressing datasets is in the form of network structures over the entity sets $\mathcal{E}_1$ and $\mathcal{E}_2$. For example, in a recommendation engine, such a structure might be available in the form of a social network of users. In addition, a taxonomy might also be available for the items. Such network structures encode important preference characteristics of different entities, thereby influencing the affinities between them [15]. Systematically accounting for such networks can greatly improve the estimation of missing affinities. Consider an example of a movie recommendation engine with additional network information; users that are *mutual friends* in a given social network often have similar tastes and preferences for some movies. Similarly, movies *connected* by same actors tend to solicit similar affinity behaviors from the users. Hence, leveraging network structures over the two sets of entities can greatly improve the prediction of missing affinities. In this

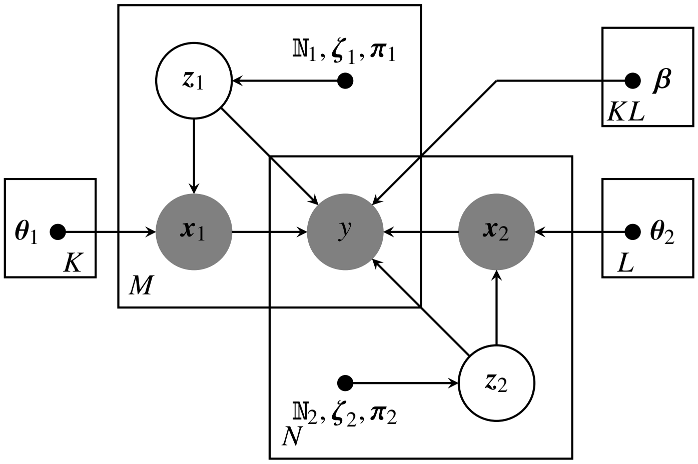

Figure 3: Graphical model for Neighborhood Aware Bayesian Affinity Estimation

section, we incorporate such network structures into the SABAE framework. The resulting 'Neighborhood Aware Bayesian Affinity Estimation' (NA-BAE) framework greatly improves the prediction accuracy both for warm and cold-start scenarios.

Let $\mathbb{N}_1 = \{\mathcal{N}_{1m}, \boldsymbol{\zeta}_1\}, [m]_1^M$ represent a weighted network structure over the entities in the set $\mathcal{E}_1$. $\mathcal{N}_{1m}$ is the set of entities $e_{1i} \in \mathcal{E}_1$ that lie in the neighborhood of entity e_{1m}. In addition, the strength of the neighborhood relation is encoded by a set of weights $\boldsymbol{\zeta}_{1m} = \{\zeta_{1mi}\}, i \in \mathcal{N}_{1m}$. Similarly, let $\mathbb{N}_2 = \{\mathcal{N}_{2n}, \boldsymbol{\zeta}_2\}, [n]_1^N$ be a network structure over the entities in the set $\mathcal{E}_2$. The neighborhood along with the associated weights for an entity $e_{2n} \in \mathcal{E}_2$ are denoted respectively, by $\mathcal{N}_{2n}$ and $\boldsymbol{\zeta}_{2n} = \{\zeta_{1nj}\}, j \in \mathcal{N}_{2n}$. Assuming that the network structures along with the link strengths are known *apriori*, we incorporate the neighborhood structures in the form of separate Markov random field priors [29] over cluster assignment latent variables $\mathcal{Z}_1$ and $\mathcal{Z}_2$ (see figure 3 for a graphical model representation). The joint prior distribution of the latent cluster assignment variables is then given by:

$$p(\mathcal{Z}_1|\mathbb{N}_1, \boldsymbol{\zeta}_1, \boldsymbol{\pi}_1) \propto \prod_m \exp\left(\sum_{i\in\mathcal{N}_{1m}} \zeta_{1mi}\mathbb{1}_{\{z_{1m}=z_{1i}\}} + \log \boldsymbol{\pi}_{1z_{1m}}\right) \tag{29}$$

$$p(\mathcal{Z}_2|\mathbb{N}_2, \boldsymbol{\zeta}_2, \boldsymbol{\pi}_2) \propto \prod_n \exp\left(\sum_{j\in\mathcal{N}_{2n}} \zeta_{2nj}\mathbb{1}_{\{z_{2n}=z_{2j}\}} + \log \boldsymbol{\pi}_{2z_{2n}}\right) \tag{30}$$

where $\mathbb{1}_{\{z_{1m}=z_{1i}\}}$ is an indicator random variable that assumes a value 1 if the clustering assignments z_{1m}, z_{1i} match. Similarly, the indicator random variable $\mathbb{1}_{\{z_{2l}=z_{2j}\}}$ assumes a value 1 if the entities (e_{2n}, e_{2j}) are assigned to the same cluster. The MRF priors defined above capture an essential neighborhood property by assigning the two neighboring entities e_{1m} and e_{1i} (e_{2n} and e_{2j}) to the same cluster with a prior probability proportional to the strength of neighborhood weight, $e^{\zeta_{1mi}}$ ($e^{\zeta_{2nj}}$). Hence, the affinity relationships for a particular entity is influenced by its neighborhood in the given network structure. The mixing coefficients $\boldsymbol{\pi}_1$ and $\boldsymbol{\pi}_2$ represent the probability of entities in $\mathcal{E}_1$ and $\mathcal{E}_2$ respectively, being assigned to a particular cluster independent of their neighborhood. This helps to capture entity level cluster assignment probabilities. The mixing coefficients can first be obtained as expectations under the Dirichlet distribution prior with learnt parameters $\boldsymbol{\alpha}_1$ and $\boldsymbol{\alpha}_2$ using the LD-AA-BAE model described in section 3.2.

The overall joint distribution over all observable and latent variables is given by:

$$p(\mathcal{Y}, \mathcal{X}_1, \mathcal{X}_2, \mathcal{Z}_1, \mathcal{Z}_2 | \boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2, \boldsymbol{\beta}, \mathbb{N}_1, \boldsymbol{\zeta}_1, \mathbb{N}_2, \boldsymbol{\zeta}_2, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2) =$$

$$p(\mathcal{Z}_1 | \mathbb{N}_1, \boldsymbol{\zeta}_1, \boldsymbol{\pi}_1) p(\mathcal{Z}_2 | \mathbb{N}_2, \boldsymbol{\zeta}_2, \boldsymbol{\pi}_2) \left(\prod_m p_{\psi_1}(\boldsymbol{x}_{1m} | \boldsymbol{\theta}_{1z_{1m}}) \right) \left(\prod_n p_{\psi_2}(\boldsymbol{x}_{2n} | \boldsymbol{\theta}_{2z_{2n}}) \right) \left(\prod_{m,n} p_{\psi_{\mathcal{Y}}}(y_{mn} | \boldsymbol{\beta}^{\dagger}_{z_{1m}z_{2n}} \boldsymbol{x}_{mn}) \right)$$

The marginal distribution of the observed affinities and the attributes is obtained by integrating out the latent variables:

$$p(\mathcal{Y}_{\text{obs}}, \mathcal{X}_1, \mathcal{X}_2 | \boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2, \boldsymbol{\beta}, \mathbb{N}_1, \boldsymbol{\zeta}_1, \mathbb{N}_2, \boldsymbol{\zeta}_2, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2) =$$

$$\int \sum_{\mathcal{Z}_1} \sum_{\mathcal{Z}_2} p(\mathcal{Y}, \mathcal{X}_1, \mathcal{X}_2, \mathcal{Z}_1, \mathcal{Z}_2 | \boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2, \boldsymbol{\beta}, \mathbb{N}_1, \boldsymbol{\zeta}_1, \mathbb{N}_2, \boldsymbol{\zeta}_2, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2) d\mathcal{Y}_{\text{unobs}}$$

The free model parameters $(\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2, \boldsymbol{\beta})$ can be estimated by maximizing the observed log-likelihood. However, due to the correlations induced by the MRF priors, computation of the observed likelihood is intractable and requires a marginalization over all configurations of $\mathcal{Z}_1$ and $\mathcal{Z}_2$ (exponential in the size of largest clique in the network structures).

4.1 Inference and Learning

To overcome the intractability of directly maximizing the observed log-likelihood, we construct tractable lower bounds to the likelihood using a mean field approximation to the true posterior distribution of the latent variables. The lower bound can then be maximized with respect to the free model parameters to get a better estimate of their values.

We approximate the true posterior distribution over the latent variables by a fully factorized distribution of the following form:

$$q(\mathcal{Y}_{\text{unobs}}, \mathcal{Z}_1, \mathcal{Z}_2) = \left(\prod_{\substack{m,n \\ y_{mn} \in \mathcal{Y}_{\text{unobs}}}} q(y_{mn}) \right) \left(\prod_m q(z_{1m}) \right) \left(\prod_n q(z_{2n}) \right) \tag{31}$$

Under this distribution over the latent variables, the observed log-likelihood is bounded from below as follows:

$$\log p(\mathcal{Y}_{\text{obs}}, \mathcal{X}_1, \mathcal{X}_2 | \boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2, \boldsymbol{\beta}, \mathbb{N}_1, \boldsymbol{\zeta}_1, \mathbb{N}_2, \boldsymbol{\zeta}_2, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2) \geq$$

$$H[q] + \mathbb{E}_q[\log p(\mathcal{Y}, \mathcal{X}_1, \mathcal{X}_2, \mathcal{Z}_1, \mathcal{Z}_2 | \boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2, \boldsymbol{\beta}, \mathbb{N}_1, \boldsymbol{\zeta}_1, \mathbb{N}_2, \boldsymbol{\zeta}_2, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2)]$$

Maximizing this lower bound over all possible factorized distributions of the form in (31), the distribution corresponding to the tightest lower bound is then given as follows (see section 3.1):

$$q^*(\mathcal{Y}_{\text{unobs}}, \mathcal{Z}_1, \mathcal{Z}_2) = \left(\prod_{\substack{m,n \\ y_{mn} \in \mathcal{Y}_{\text{unobs}}}} q^*_{\psi_{\mathcal{Y}}}(y_{mn} | \phi_{mn}) \right) \left(\prod_m q^*(z_{1m} | r_{1m}) \right) \left(\prod_n q^*(z_{2n} | r_{2n}) \right)$$

The variational distribution $q^*_{\psi_{\mathcal{Y}}}(y_{mn}|\phi_{mn})$ is an exponential family distribution having same form as the one assumed for the affinities while ϕ_{mn} is the natural parameter of the distribution. Similarly, variational distributions over the cluster assignments $q^*(z_{1m}|r_{1m})$ and $q^*(z_{2n}|r_{2n})$ follow discrete distribution

Algorithm 3 Learn NA-BAE

```
Input: $\mathcal{Y}_{\text{obs}}, \mathcal{X}_1, \mathcal{X}_2, \mathbb{N}_1, \boldsymbol{\zeta}_1, \mathbb{N}_2, \boldsymbol{\zeta}_2, K, L$
Output: $\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2, \boldsymbol{\beta}$
        $[m]_1^M, [n]_1^N, [k]_1^K, [l]_1^L$

    Step 0a: Initialize $\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2, \boldsymbol{\beta}, \boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2$ using algorithm 2
    Step 0b: Assign $\boldsymbol{\pi}_{1k} = \frac{\alpha_{1k}}{\sum_{k'=1}^{K} \alpha_{1k'}}$
    Step 0c: Assign $\boldsymbol{\pi}_{2l} = \frac{\alpha_{2l}}{\sum_{l'=1}^{L} \alpha_{2l'}}$
    Until Convergence
        Step 1: E-Step
            Step 1a: Initialize $r_{1mk}, r_{2nl}$
            Until Convergence
                Step 1b: Update $\phi_{mn}$ using equation (32)
                Step 1c: Update $r_{1mk}$ using equation (33)
                Step 1d: Update $r_{2nl}$ using equation (34)
        Step 2: M-Step
            Step 2a: Update $\boldsymbol{\theta}_{1k}$ using equation (35)
            Step 2b: Update $\boldsymbol{\theta}_{2l}$ using equation (36)
            Step 2c: Update $\boldsymbol{\beta}_{kl}$ using equation (37)
```

over K and L clusters with parameters r_{1m} and r_{2n} respectively. The variational parameters $(\phi_{mn}, r_{1m}, r_{2n})$ are then given by the following equations (see Appendix B for details):

$$\phi_{mn} = \sum_{k=1}^{K} \sum_{l=1}^{L} r_{1mk} r_{2nl} \left(\boldsymbol{\beta}_{kl}^{\dagger} \boldsymbol{x}_{mn} \right) \tag{32}$$

$$r_{1mk} \propto \exp\Bigg(\log p_{\psi_1}(\boldsymbol{x}_{1m} | \boldsymbol{\theta}_{1k}) + \sum_{i \in \mathcal{N}_{1m}} \zeta_{1mi} r_{1ik} + \log \boldsymbol{\pi}_{1k} \tag{33}$$

$$+ \sum_{n=1}^{N} \sum_{l=1}^{L} r_{2nl} \left(w_{mn} \log p_{\psi_y}(y_{mn} | \boldsymbol{\beta}_{kl}^{\dagger} \boldsymbol{x}_{mn}) + (1 - w_{mn}) \mathbb{E}_q \left[\log p_{\psi_y}(y_{mn} | \boldsymbol{\beta}_{kl}^{\dagger} \boldsymbol{x}_{mn}) \right] \right) \Bigg)$$

$$r_{2nl} \propto \exp\Bigg(\log p_{\psi_2}(\boldsymbol{x}_{2n} | \boldsymbol{\theta}_{2l}) + \sum_{j \in \mathcal{N}_{2n}} \zeta_{2jl} r_{2jl} + \log \boldsymbol{\pi}_{2l} \tag{34}$$

$$+ \sum_{m=1}^{M} \sum_{k=1}^{K} r_{1mk} \left(w_{mn} \log p_{\psi_y}(y_{mn} | \boldsymbol{\beta}_{kl}^{\dagger} \boldsymbol{x}_{mn}) + (1 - w_{mn}) \mathbb{E}_q \left[\log p_{\psi_y}(y_{mn} | \boldsymbol{\beta}_{kl}^{\dagger} \boldsymbol{x}_{mn}) \right] \right) \Bigg)$$

The coupled mean field equations for the variational parameter updates can be satisfied iteratively to obtain a lower bound on the observed log-likelihood. Note that the estimate of the posterior cluster responsibilities $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ are influenced by the MRF based priors (the terms $\sum_{i \in \mathcal{N}_m} \zeta_{1mi} r_{1ik}$ and $\sum_{j \in \mathcal{N}_n} \zeta_{2jl} r_{2jl}$ in the updates for r_{1mk} and r_{2nl}). In particular, the posterior cluster assignment probability for each entity is influenced by its neighboring entities. This helps us to capture the notion of *neighborhood similarity* wherein neighboring entities tend to have similar affinities with a high probability.

The lower bound obtained from the inference step can be then maximized with respect to the free model parameters. Setting partial derivatives of the bound with respect to the model parameters to zero, the following updates are obtained (see Appendix B for details):

$$\boldsymbol{\theta}_{1k} = \nabla\psi_1^{-1}\left(\frac{\sum_{m=1}^{M} r_{1mk}\boldsymbol{x}_{1m}}{\sum_{m=1}^{M} r_{1mk}}\right) \tag{35}$$

$$\boldsymbol{\theta}_{2l} = \nabla\psi_2^{-1}\left(\frac{\sum_{n=1}^{N} r_{2nl}\boldsymbol{x}_{2n}}{\sum_{n=1}^{N} r_{2nl}}\right) \tag{36}$$

$$\boldsymbol{\beta}_{kl} = \arg\max_{\beta\in\mathbb{R}^D} \sum_{m=1}^{M}\sum_{n=1}^{N} r_{1mk}r_{2nl}\left[\left\langle \left(w_{mn}y_{mn} + (1-w_{mn})\nabla\psi_{\mathcal{Y}}(\phi_{mn})\right), \boldsymbol{\beta}^{\dagger}\boldsymbol{x}_{mn}\right\rangle - \psi_{\mathcal{Y}}\left(\boldsymbol{\beta}^{\dagger}\boldsymbol{x}_{mn}\right)\right] \tag{37}$$

An EM style algorithm can then be derieved wherein E-step a tight lower bound is constructed on the observed likelihood by iteratively satisfying the mean field equations (32) through (34) for a fixed value of the free model parameters. The optimized lower bound is then maximized in the subsequent M-step to get an improved estimate of the model parameters. The resulting EM algorithm is given in algorithm 3.

5 Factorization Aware Bayesian Affinity Estimation

The major data mining tasks associated with affinity expressing datasets are two folds, first an interpretable *understanding* of the affinity relationships between the entities and secondly, inferences about the missing affinities. While clustering based approaches yield easily interpretable results, they often suffer from a weaker prediction capability. Recently, factorization based approaches have been shown to give good performance on imputing the missing affinities ([33], [3], [8]). In this section, we embed a factorized representation of the affinity relationships into SABAE framework. The resulting factorization aware Bayesian affinity estimation (FA-BAE) framework allows us to achieve the twin goals of interpretability and superior predictive performance.

Most factorization based approaches assume that the affinity relationship between two entities is dependent on a small number of entity specific unobserved factors. If $\boldsymbol{u}_m \in \mathbb{R}^t$ are t dimensional factors associated with entity $e_{1m} \in \mathcal{E}_1$, while $\boldsymbol{v}_n \in \mathbb{R}^t$ are the factors corresponding to $e_{2n} \in \mathcal{E}_2$, then the expected value of the affinity y_{mn} between the two entities is modeled as a linear combination of the two factors [3]:

$$\mathbb{E}[y_{mn}] = \boldsymbol{u}_m^{\dagger}\boldsymbol{v}_n$$

We embed the above factorized representation into the SABAE framework by enforcing the affinities in the same cluster to have similar factorized represenation. Specifically, we assume that the individual factors are sampled from a normal distribution with co-cluster specific parameters. Let $\mathcal{U} = \{\boldsymbol{u}_m\}, [m]_1^M$ and $\mathcal{V} = \{\boldsymbol{v}_n\}, [n]_1^N$ denote the factors associated with the entities in sets $\mathcal{E}_1$ and $\mathcal{E}_2$ respectively, then the conditional distribution of the factors can be expressed as follows:

$$p(\mathcal{U}|\mathcal{Z}_1) = \prod_{m=1}^{M} N(\boldsymbol{u}_m|\boldsymbol{\mu}_{1z_{1m}}, \boldsymbol{\Sigma}_{1z_{1m}})$$

$$p(\mathcal{V}|\mathcal{Z}_2) = \prod_{n=1}^{N} N(\boldsymbol{v}_n|\boldsymbol{\mu}_{2z_{2n}}, \boldsymbol{\Sigma}_{2z_{2n}})$$

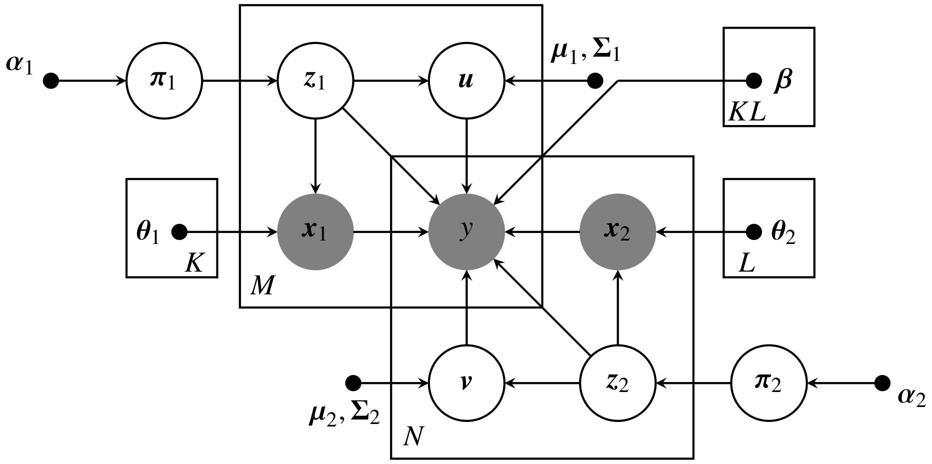

Figure 4: Graphical model for Factorization Aware Bayesian Affinity Estimation

where $N(\cdot|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. To account for the attribute information, we model the expected value of an affinity as:

$$\mathbb{E}[y_{mn}] = \boldsymbol{u}_m^\dagger \boldsymbol{v}_n + \boldsymbol{\beta}_{z_{1m}z_{2n}}^\dagger \boldsymbol{x}_{mn}$$

For simplicity in inference, we further assume the conditional distribution of the affinities conditioned on the cluster assignment variables to be a Gaussian distribution. Hence, the complete likelihood over all observed and latent variables is then given by:

$$p(\mathcal{Y}, \mathcal{X}_1, \mathcal{X}_2, \mathcal{Z}_1, \mathcal{Z}_2, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \mathcal{U}, \mathcal{V}|\alpha_1, \alpha_2, \boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2, \boldsymbol{\beta}, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2) =$$

$$p(\boldsymbol{\pi}_1|\alpha_1)p(\boldsymbol{\pi}_2|\alpha_2)\left(\prod_m p(z_{1m}|\boldsymbol{\pi}_1)p_{\psi_1}(\boldsymbol{x}_{1m}|\boldsymbol{\theta}_{1z_{1m}})N(\boldsymbol{u}_m|\boldsymbol{\mu}_{1_{z_{1m}}}, \boldsymbol{\Sigma}_{1z_{1m}})\right)\left(\prod_n p(z_{2n}|\boldsymbol{\pi}_2)p_{\psi_2}(\boldsymbol{x}_{2n}|\boldsymbol{\theta}_{2z_{2n}})N(\boldsymbol{v}_n|\boldsymbol{\mu}_{2_{z_{2n}}}, \boldsymbol{\Sigma}_{2z_{2n}})\right)$$

$$\left(\prod_{m,n} N(y_{mn}|\boldsymbol{u}_m^\dagger \boldsymbol{v}_n + \boldsymbol{\beta}_{z_{1m}z_{2n}}^\dagger \boldsymbol{x}_{mn}, \sigma^2_{z_{1m}z_{2n}})\right)$$

5.1 Inference and Learning

The free model parameters $(\alpha_1, \alpha_2, \boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2, \boldsymbol{\beta}, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2, \sigma^2)$ can be learnt by maximizing the observed log-likelihood using an EM algorithm. However, due to the sharing of the attributes and the factors for affinities associated with an entity, computation of the observed log-likelihood is intractable. To overcome the intractability of directly maximizing the observed log-likelihood, we construct tractable lower bounds using a mean field approximation to the true posterior distribution of the latent variables.

Following the analysis of section 3.1, we approximate the true posterior distribution by a fully factorized distribution of the following form:

$$q^*(\mathcal{Y}_{\text{unobs}}, \mathcal{Z}_1, \mathcal{Z}_2, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \mathcal{U}, \mathcal{V}) = \tag{38}$$

$$q^*(\boldsymbol{\pi}_1|\boldsymbol{\gamma}_1)q^*(\boldsymbol{\pi}_2|\boldsymbol{\gamma}_2)\left(\prod_{\substack{m,n \\ y_{mn}\in\mathcal{Y}_{\text{unobs}}}} q^*(y_{mn}|\vartheta_{mn}, \varsigma_{mn})\right)\left(\prod_m q^*(z_{1m}|\boldsymbol{r}_{1m})q^*(\boldsymbol{u}_m|\boldsymbol{\rho}_{1m}, \boldsymbol{\Lambda}_{1m})\right)\left(\prod_n q^*(z_{2n}|\boldsymbol{r}_{2n})q^*(\boldsymbol{v}_n|\boldsymbol{\rho}_{2n}, \boldsymbol{\Lambda}_{2n})\right)$$

where $(\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2, \vartheta_{mn}, \varsigma_{mn}, \boldsymbol{r}_{1m}, \boldsymbol{r}_{2n}, \boldsymbol{\rho}_{1m}, \boldsymbol{\rho}_{2n}, \boldsymbol{\Lambda}_{1m}, \boldsymbol{\Lambda}_{2n})$ are the variational parameters. The variational parameters corresponding to the tightest lowest bound with respect to the factorized distribution in (38) can then be expressed by the following coupled mean field equations:

The variational distribution $q^*(y_{mn}|\vartheta_{mn}, \varsigma_{mn})$ is a Gaussian distribution with mean ϑ_{mn} and variance ς_{mn} such that,

$$\vartheta_{mn} = \boldsymbol{\rho}_{1m}^{\dagger}\boldsymbol{\rho}_{2n} + \frac{\sum_{k,l=1}^{K,L} \frac{r_{1mk}r_{2nl}}{\sigma_{kl}^2}\boldsymbol{\beta}_{kl}^{\dagger}\boldsymbol{x}_{mn}}{\sum_{k,l=1}^{K,L} \frac{r_{1mk}r_{2nl}}{\sigma_{kl}^2}} \tag{39}$$

$$\varsigma_{mn}^2 = \frac{1}{\sum_{k,l=1}^{K,L} \frac{r_{1mk}r_{2nl}}{\sigma_{kl}^2}} \tag{40}$$

Distributions $q^*(\boldsymbol{\pi}_1|\boldsymbol{\gamma}_1)$ and $q^*(\boldsymbol{\pi}_2|\boldsymbol{\gamma}_2)$ are K and L dimensional Dirichlet distributions with parameters $\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2$ respectively.

$$\gamma_{1k} = \sum_{m=1}^{M} r_{1mk} + \alpha_{1k} \tag{41}$$

$$\gamma_{2l} = \sum_{n=1}^{N} r_{2nl} + \alpha_{2l} \tag{42}$$

The variational distributions $q^*(\boldsymbol{u}_m|\boldsymbol{\rho}_{1m}, \boldsymbol{\Lambda}_{1m})$ and $q^*(\boldsymbol{v}_n|\boldsymbol{\rho}_{2n}, \boldsymbol{\Lambda}_{2n})$ corresponding to the optimal lower bound are multivariate Gaussian distributions with means $\boldsymbol{\rho}_{1m}, \boldsymbol{\rho}_{2n}$ and the precision matrices $\boldsymbol{\Lambda}_{1m}$ and $\boldsymbol{\Lambda}_{2n}$ respectively. The mean field equations for the parameters of these multivariate Gaussian distributions are then given by:

$$\boldsymbol{\rho}_{1m} = \sum_{k=1}^{K} r_{1mk}\left(\boldsymbol{\Sigma}_{1k}\boldsymbol{\mu}_{1k} + \sum_{n=1}^{N}\sum_{l=1}^{L}\frac{r_{2nl}\boldsymbol{\rho}_{2n}}{\sigma_{kl}^2}(y_{mn} - \boldsymbol{\beta}_{kl}^{\dagger}\boldsymbol{x}_{mn})\right) \tag{43}$$

$$\Lambda_{1m} = \sum_{k=1}^{K} r_{1mk}\left(\boldsymbol{\Sigma}_{1k} + \sum_{n=1}^{N}\sum_{l=1}^{L}\frac{r_{2nl}}{\sigma_{kl}^2}\left(\Lambda_{2n}^{-1} + \boldsymbol{\rho}_{2n}\boldsymbol{\rho}_{2n}^{\dagger}\right)\right) \tag{44}$$

$$\boldsymbol{\rho}_{2n} = \sum_{l=1}^{L} r_{2nl}\left(\boldsymbol{\Sigma}_{2l}\boldsymbol{\mu}_{2l} + \sum_{m=1}^{M}\sum_{k=1}^{K}\frac{r_{1mk}\boldsymbol{\rho}_{1m}}{\sigma_{kl}^2}(y_{mn} - \boldsymbol{\beta}_{kl}^{\dagger}\boldsymbol{x}_{mn})\right) \tag{45}$$

$$\Lambda_{2n} = \sum_{l=1}^{L} r_{2nl}\left(\boldsymbol{\Sigma}_{2l} + \sum_{m=1}^{M}\sum_{k=1}^{K}\frac{r_{1mk}}{\sigma_{kl}^2}\left(\Lambda_{1m}^{-1} + \boldsymbol{\rho}_{1m}\boldsymbol{\rho}_{1m}^{\dagger}\right)\right) \tag{46}$$

Finally, the optimal variational distributions for the cluster assignment latent variables $z_{1m} \in \mathcal{Z}_1$ and $z_{2n} \in \mathcal{Z}_2$ are K and L dimensional discrete distributions with parameters $\boldsymbol{r}_{1m}$ and $\boldsymbol{r}_{2n}$ respectively. The updates for the individual parameters is then obtained by the following equations:

$$r_{1mk} \propto \exp\Bigg(\log p_{\psi_1}(\boldsymbol{x}_{1m}|\boldsymbol{\theta}_{1k}) + \Psi(\gamma_{1k}) - \frac{1}{2}\Big[(\boldsymbol{\rho}_{1m} - \boldsymbol{\mu}_{1k})^{\dagger}\boldsymbol{\Sigma}_{1k}(\boldsymbol{\rho}_{1m} - \boldsymbol{\mu}_{1k}) + \mathrm{Tr}(\boldsymbol{\Sigma}_{1k}^{-1}\Lambda_{1m}^{-1}) + \log\boldsymbol{\Sigma}_{1k}\Big] +$$

$$\sum_{n=1}^{N}\sum_{l=1}^{L}\frac{-r_{2nl}}{2\sigma_{kl}^2}\Big[w_{mn}y_{mn}^2 + (1 - w_{mn})(\varsigma_{mn}^2 + \vartheta_{mn}^2) - 2(w_{mn}y_{mn} + (1 - w_{mn})\vartheta_{mn})(\boldsymbol{\rho}_{1m}^{\dagger}\boldsymbol{\rho}_{2n} + \boldsymbol{\beta}_{kl}^{\dagger}\boldsymbol{x}_{mn}) +$$

$$\mathrm{Tr}(\mathbb{E}_q[\boldsymbol{u}_m\boldsymbol{u}_m^\dagger]\mathbb{E}_q[\boldsymbol{v}_n\boldsymbol{v}_n^\dagger]) + (\boldsymbol{\beta}_{kl}\boldsymbol{x}_{mn})^2 + 2(\boldsymbol{\rho}_{1m}^\dagger\boldsymbol{\rho}_{2n})(\boldsymbol{\beta}_{kl}\boldsymbol{x}_{mn}) + \sigma_{kl}^2\log\sigma_{kl}\Big]\Big) \tag{47}$$

$$r_{2nl} \propto \exp\Bigg(\log p_{\psi_2}(\boldsymbol{x}_{2n}|\boldsymbol{\theta}_{2l}) + \Psi(\gamma_{2l}) - \frac{1}{2}\Big[(\boldsymbol{\rho}_{2n}-\boldsymbol{\mu}_{2l})^\dagger\boldsymbol{\Sigma}_{2l}(\boldsymbol{\rho}_{2n}-\boldsymbol{\mu}_{2l}) + \mathrm{Tr}(\boldsymbol{\Sigma}_{2l}^{-1}\Lambda_{2n}^{-1}) + \log\boldsymbol{\Sigma}_{2l}\Big] +$$

$$\sum_{m=1}^{M}\sum_{k=1}^{K}\frac{-r_{1mk}}{2\sigma_{kl}^2}\Big[w_{mn}y_{mn}^2 + (1-w_{mn})(\varsigma_{mn}^2+\vartheta_{mn}^2) - 2(w_{mn}y_{mn} + (1-w_{mn})\vartheta_{mn})(\boldsymbol{\rho}_{1m}^\dagger\boldsymbol{\rho}_{2n} + \boldsymbol{\beta}_{kl}^\dagger\boldsymbol{x}_{mn}) +$$

$$\mathrm{Tr}(\mathbb{E}_q[\boldsymbol{u}_m\boldsymbol{u}_m^\dagger]\mathbb{E}_q[\boldsymbol{v}_n\boldsymbol{v}_n^\dagger]) + (\boldsymbol{\beta}_{kl}\boldsymbol{x}_{mn})^2 + 2(\boldsymbol{\rho}_{1m}^\dagger\boldsymbol{\rho}_{2n})(\boldsymbol{\beta}_{kl}\boldsymbol{x}_{mn}) + \sigma_{kl}^2\log\sigma_{kl}\Big]\Big) \tag{48}$$

where $\mathrm{Tr}(\cdot)$ denotes the trace of a square matrix and $\Psi(\cdot)$ is the digamma function. The expectations $\mathbb{E}[\boldsymbol{u}_m\boldsymbol{u}_m^\dagger]$ and $\mathbb{E}[\boldsymbol{v}_n\boldsymbol{v}_n^\dagger]$ are the second moments of the factors $\boldsymbol{u}_m$ and $\boldsymbol{v}_n$ under the variational multivariate Gaussian distributions with mean and precision matrices given by equations (43) through (46). Note that by utilizing the weights w_{mn} associated with each rating y_{mn}, we ensure that for any missing rating (i.e. $w_{mn} = 0$), the expressions involving the ratings are replaced by appropriate expectations under the variational distribution $q^*(y_{mn}|\varsigma_{mn}, \vartheta_{mn})$.

The mean field equations derived above can be satisfied iteratively to obtain a lower bound on the observed log-likelihood. The bound can then be used as a pseudo likelihood for parameter estimation. Specifically, maximizing the lower bound with respect to the free model parameters we obtain the following update equations:

$$\boldsymbol{\theta}_{1k} = \nabla\psi_1^{-1}\left(\frac{\sum_{m=1}^{M} r_{1mk}\boldsymbol{x}_{1m}}{\sum_{m=1}^{M} r_{1mk}}\right) \tag{49}$$

$$\boldsymbol{\theta}_{2l} = \nabla\psi_2^{-1}\left(\frac{\sum_{n=1}^{N} r_{2nl}\boldsymbol{x}_{2n}}{\sum_{n=1}^{N} r_{2nl}}\right) \tag{50}$$

The updates for cluster specific parameters for the factors $(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2)$ can be obtained in closed form expressions:

$$\boldsymbol{\mu}_{1k} = \frac{\sum_{m=1}^{M} r_{1mk}\boldsymbol{u}_m}{\sum_{m=1}^{M} r_{1mk}} \tag{51}$$

$$\boldsymbol{\mu}_{2l} = \frac{\sum_{n=1}^{N} r_{2nl}\boldsymbol{v}_n}{\sum_{n=1}^{N} r_{2nl}} \tag{52}$$

$$\boldsymbol{\Sigma}_{1k} = \frac{\sum_{m=1}^{M} r_{1mk}\left[\boldsymbol{\Lambda}_{1m}^{-1} + (\boldsymbol{\rho}_{1m}-\boldsymbol{\mu}_{1k})(\boldsymbol{\rho}_{1m}-\boldsymbol{\mu}_{1k})^\dagger\right]}{\sum_{m=1}^{M} r_{1mk}} \tag{53}$$

$$\boldsymbol{\Sigma}_{2l} = \frac{\sum_{n=1}^{N} r_{2nl}\left[\boldsymbol{\Lambda}_{2n}^{-1} + (\boldsymbol{\rho}_{2n}-\boldsymbol{\mu}_{2l})(\boldsymbol{\rho}_{2n}-\boldsymbol{\mu}_{2l})^\dagger\right]}{\sum_{n=1}^{N} r_{2nl}} \tag{54}$$

A Gaussian distribution assumption over the affinities also results in a closed form updates for the glm coefficients $\boldsymbol{\beta}_{kl}$. In fact, the updates are a solution to a weighted least squares problem for the covariates $\boldsymbol{x}_{mn}$ over the residual affinities $(y_{mn} - \boldsymbol{\rho}_{1m}^\dagger\boldsymbol{\rho}_{2n})$. Note that the missing affinities are replaced by their expectations under the variational distribution. Finally, the weights are the co-cluster weights $r_{1mk}r_{2nl}$.

$$\boldsymbol{\beta}_{kl} = \left[\sum_{m,n=1}^{M,N} r_{1mk}r_{2nl}\boldsymbol{x}_{mn}\boldsymbol{x}_{mn}\right]^{-1}\left[\sum_{m,n=1}^{M,N}(w_{mn}y_{mn} + (1-w_{mn})\vartheta_{mn} - \boldsymbol{\rho}_{1m}^\dagger\boldsymbol{\rho}_{2n})\right] \tag{55}$$

Algorithm 4 Learn FA-BAE

```
Input: $\mathcal{Y}_{\text{obs}}, \mathcal{X}_1, \mathcal{X}_2, K, L$
Output: $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2, \boldsymbol{\beta}, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2, \sigma^2_{kl}$
        $[m]_1^M, [n]_1^N, [k]_1^K, [l]_1^L$

  Step 0: Initialize $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2, \boldsymbol{\beta}, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2, \sigma^2_{kl}$
  Until Convergence
      Step 1: E-Step
          Step 1a: Initialize $r_{1mk}, r_{2nl}, \boldsymbol{\rho}_{1m}, \boldsymbol{\rho}_{2n}$
          Until Convergence
              Step 1b: Update ($\vartheta_{mn}$, $\varsigma_{mn}$) using equations (39) and (40)
              Step 1c: Update ($\gamma_{1k}$, $\gamma_{2l}$) using equations (41) and (42)
              Step 1d: Update ($\boldsymbol{\rho}_{1m}$, $\boldsymbol{\rho}_{2n}$) using equations (43) and (45)
              Step 1e: Update ($\boldsymbol{\Lambda}_{1m}$, $\boldsymbol{\Lambda}_{2n}$) using equations (44) and (46)
              Step 1f: Update ($r_{1mk}$, $r_{2nl}$) using equations (47) and (48)
      Step 2: M-Step
          Step 2a: Update ($\boldsymbol{\theta}_{1k}$, $\boldsymbol{\theta}_{2l}$) using equations (49) and (50)
          Step 2b: Update ($\boldsymbol{\mu}_{1k}$, $\boldsymbol{\mu}_{2l}$) using equations (51) and (52)
          Step 2c: Update ($\boldsymbol{\Sigma}_{1k}$, $\boldsymbol{\Sigma}_{2l}$) using equations (53) and (54)
          Step 2d: Update $\boldsymbol{\beta}_{kl}$ using equation (55)
          Step 2e: Update $\sigma^2_{kl}$ using equation (56)
          Step 2f: Update ($\boldsymbol{\alpha}_1$, $\boldsymbol{\alpha}_2$) using equations (57) and (58)
```

Finally, the updates for the co-cluster variances σ_{kl} and the parameters of the Dirichlet distribution priors $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2$ are given by:

$$\sigma^2_{kl} = \frac{\sum_{m,n=1}^{M,N} r_{1mk} r_{2nl} \mathbb{E}_q[(y_{mn} - \boldsymbol{u}_m^\dagger \boldsymbol{v}_n - \boldsymbol{\beta}_{kl}^\dagger \boldsymbol{x}_{mn})^2]}{\sum_{m,n=1}^{M,N} r_{1mk} r_{2nl}} \tag{56}$$

$$\boldsymbol{\alpha}_1 = \operatorname*{arg\,max}_{\boldsymbol{\alpha}_1 \in \mathbb{R}_{++}^K} \left(\log \frac{\Gamma(\sum_{k=1}^K \alpha_{1k})}{\prod_{k=1}^K \Gamma(\alpha_{1k})} + \sum_{k=1}^K \left(\alpha_{1k} + \sum_{m=1}^M r_{1mk} - 1 \right) \left(\Psi(\gamma_{1k}) - \Psi\left(\sum_{k'=1}^K \gamma_{1k'} \right) \right) \right) \tag{57}$$

$$\boldsymbol{\alpha}_2 = \operatorname*{arg\,max}_{\boldsymbol{\alpha}_2 \in \mathbb{R}_{++}^L} \left(\log \frac{\Gamma(\sum_{l=1}^L \alpha_{2l})}{\prod_{l=1}^L \Gamma(\alpha_{2l})} + \sum_{l=1}^L \left(\alpha_{2l} + \sum_{n=1}^N r_{2nl} - 1 \right) \left(\Psi(\gamma_{2l}) - \Psi\left(\sum_{l'=1}^L \gamma_{2l'} \right) \right) \right) \tag{58}$$

The resulting variational EM algorithm for learning the parameters of the model is given in algorithm 4.

5.2 Neighborhood Aware Factorization

A factorized representation can easily be embedded into the NA-BAE (see section 4) framework, by enforcing the affinities within the same co-clusters, now driven by the network structure sensitive prior on the cluster assignment variables $\mathcal{Z}_1$ and $\mathcal{Z}_2$, to have a similar factorized representation. Hence, following treatment introduced for factorized aware Bayesian affinity estimation, the individual factors $\boldsymbol{u}_m$ and $\boldsymbol{v}_n$ are sampled from a normal distribution with co-cluster specific parameters.

The complete likelihood for observed and latent variables is then given by:

$$p(\mathcal{Y}, \mathcal{X}_1, \mathcal{X}_2, \mathcal{Z}_1, \mathcal{Z}_2, \mathcal{U}, \mathcal{V} | \boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2, \boldsymbol{\beta}, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2, \mathbb{N}_1, \boldsymbol{\zeta}_1, \mathbb{N}_2, \boldsymbol{\zeta}_2, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2) =$$

$$p(\mathcal{Z}_1 | \mathbb{N}_1, \boldsymbol{\zeta}_1, \boldsymbol{\pi}_1) p(\mathcal{Z}_2 | \mathbb{N}_2, \boldsymbol{\zeta}_2, \boldsymbol{\pi}_2) \left(\prod_m p_{\psi_1}(\boldsymbol{x}_{1m} | \boldsymbol{\theta}_{1z_{1m}}) N(\boldsymbol{u}_m | \boldsymbol{\mu}_{1z_{1m}}, \boldsymbol{\Sigma}_{1z_{1m}}) \right) \left(\prod_n p_{\psi_2}(\boldsymbol{x}_{2n} | \boldsymbol{\theta}_{2z_{2n}}) N(\boldsymbol{v}_n | \boldsymbol{\mu}_{2z_{2n}}, \boldsymbol{\Sigma}_{2z_{2n}}) \right)$$

$$\left(\prod_{m,n} N(y_{mn}|\boldsymbol{u}_m^\dagger \boldsymbol{v}_n + \boldsymbol{\beta}^\dagger_{z_{1m}z_{2n}}\boldsymbol{x}_{mn}, \sigma^2_{z_{1m}z_{2n}})\right)$$

where similar to the case of NA-BAE framework, the mixing coefficients $\boldsymbol{\pi}_1$ and $\boldsymbol{\pi}_2$ can be first learnt using the FA-BAE framework without the network structure information. Similar to FA-BAE, inference and parameter estimation can be done using a mean field approximation to the true posterior distribution over the latent variables.

$$q(\mathcal{Y}_{\text{unobs}}, \mathcal{Z}_1, \mathcal{Z}_2, \mathcal{U}, \mathcal{V}) = \tag{59}$$

$$\left(\prod_{\substack{m,n \\ y_{mn}\in\mathcal{Y}_{\text{unobs}}}} q(y_{mn}|\vartheta_{mn}, \varsigma_{mn})\right)\left(\prod_m q(z_{1m}|\boldsymbol{r}_{1m})q(\boldsymbol{u}_m|\boldsymbol{\rho}_{1m}, \boldsymbol{\Lambda}_{1m})\right)\left(\prod_n q(z_{2n}|\boldsymbol{r}_{2n})q(\boldsymbol{v}_n|\boldsymbol{\rho}_{2n}, \boldsymbol{\Lambda}_{2n})\right)$$

Since, the neighborhood aware factorization differs from the FA-BAE framework only in terms of the prior distribution over the cluster assignment variables, the mean field equations remain unchanged for all the variational parameters except for the cluster assignment parameters r_{1mk} and r_{2nl}, where the terms involving the Dirichlet distribution variational parameters for the mixing coefficients are replaced by the neighborhood terms from the MRF prior. Also, since the mixing coefficients are first learnt using the FA-BAE framework, no variational distribution is assumed in the factorized variational distribution. The mean field equations for the cluster assignment variational parameters is then obtained by the following equations:

$$\begin{aligned}
r_{1mk} \propto \exp\Bigg(&\log p_{\psi_1}(\boldsymbol{x}_{1m}|\boldsymbol{\theta}_{1k}) + \sum_{i\in\mathcal{N}_{1m}} \zeta_{1mi}r_{1ik} + \log\boldsymbol{\pi}_{1k} - \frac{1}{2}\left[(\boldsymbol{\rho}_{1m} - \boldsymbol{\mu}_{1k})^\dagger\boldsymbol{\Sigma}_{1k}(\boldsymbol{\rho}_{1m} - \boldsymbol{\mu}_{1k}) + \mathrm{Tr}(\boldsymbol{\Sigma}_{1k}^{-1}\Lambda_{1m}^{-1}) + \log\boldsymbol{\Sigma}_{1k}\right] + \\
&\sum_{n=1}^{N}\sum_{l=1}^{L}\frac{-r_{2nl}}{2\sigma^2_{kl}}\Big[w_{mn}y^2_{mn} + (1 - w_{mn})(\varsigma^2_{mn} + \vartheta^2_{mn}) - 2(w_{mn}y_{mn} + (1 - w_{mn})\vartheta_{mn})(\boldsymbol{\rho}^\dagger_{1m}\boldsymbol{\rho}_{2n} + \boldsymbol{\beta}^\dagger_{kl}\boldsymbol{x}_{mn}) + \\
&\mathrm{Tr}(\mathbb{E}_q[\boldsymbol{u}_m\boldsymbol{u}^\dagger_m]\mathbb{E}_q[\boldsymbol{v}_n\boldsymbol{v}^\dagger_n]) + (\boldsymbol{\beta}_{kl}\boldsymbol{x}_{mn})^2 + 2(\boldsymbol{\rho}^\dagger_{1m}\boldsymbol{\rho}_{2n})(\boldsymbol{\beta}_{kl}\boldsymbol{x}_{mn}) + \sigma^2_{kl}\log\sigma_{kl}\Big]\Bigg)
\end{aligned}\tag{60}$$

$$\begin{aligned}
r_{2nl} \propto \exp\Bigg(&\log p_{\psi_2}(\boldsymbol{x}_{2n}|\boldsymbol{\theta}_{2l}) + \sum_{j\in\mathcal{N}_{2n}} \zeta_{2nj}r_{2jl} + \log\boldsymbol{\pi}_{2l} - \frac{1}{2}\left[(\boldsymbol{\rho}_{2n} - \boldsymbol{\mu}_{2l})^\dagger\boldsymbol{\Sigma}_{2l}(\boldsymbol{\rho}_{2n} - \boldsymbol{\mu}_{2l}) + \mathrm{Tr}(\boldsymbol{\Sigma}_{2l}^{-1}\Lambda_{2n}^{-1}) + \log\boldsymbol{\Sigma}_{2l}\right] + \\
&\sum_{m=1}^{M}\sum_{k=1}^{K}\frac{-r_{1mk}}{2\sigma^2_{kl}}\Big[w_{mn}y^2_{mn} + (1 - w_{mn})(\varsigma^2_{mn} + \vartheta^2_{mn}) - 2(w_{mn}y_{mn} + (1 - w_{mn})\vartheta_{mn})(\boldsymbol{\rho}^\dagger_{1m}\boldsymbol{\rho}_{2n} + \boldsymbol{\beta}^\dagger_{kl}\boldsymbol{x}_{mn}) + \\
&\mathrm{Tr}(\mathbb{E}_q[\boldsymbol{u}_m\boldsymbol{u}^\dagger_m]\mathbb{E}_q[\boldsymbol{v}_n\boldsymbol{v}^\dagger_n]) + (\boldsymbol{\beta}_{kl}\boldsymbol{x}_{mn})^2 + 2(\boldsymbol{\rho}^\dagger_{1m}\boldsymbol{\rho}_{2n})(\boldsymbol{\beta}_{kl}\boldsymbol{x}_{mn}) + \sigma^2_{kl}\log\sigma_{kl}\Big]\Bigg)
\end{aligned}\tag{61}$$

6 Observation Aware Bayesian Affinity Estimation

In developing the SABAE framework for estimating affinities between sets of entities we have assumed that the missing affinities are missing uniformly at random (often referred to as Missing at Random (MAR) assumption. For a further explanation of the MAR assumption, see [7]). However, for most datasets describing affinity relationships, the observed affinities are recorded in a self-selecting manner. For example, in an internet-based movie recommendation engine, a user is likely to watch a movie that he or she expects to enjoy. If that expectation is fully met, the user is more likely to record the high rating

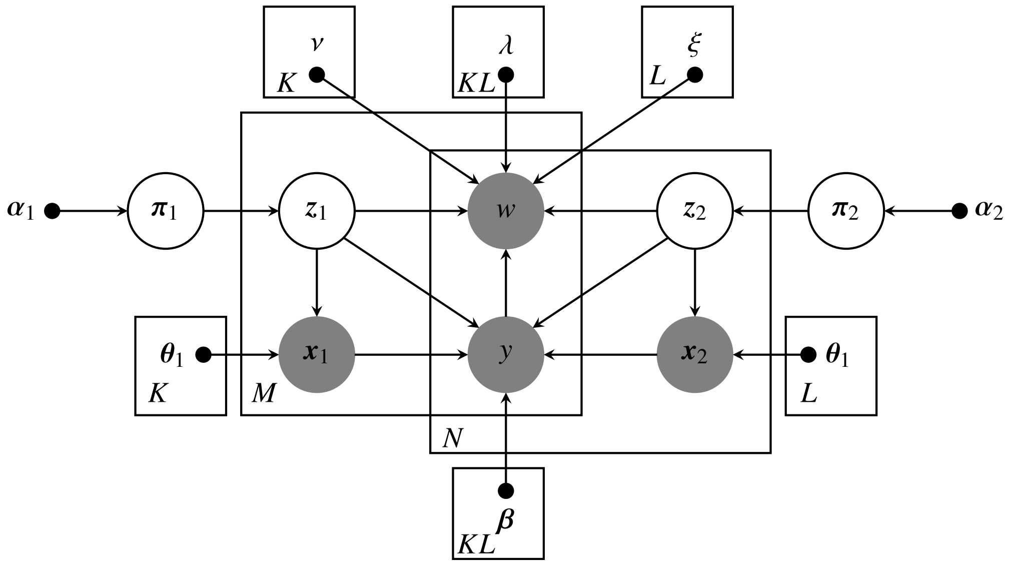

Figure 5: Graphical model for Observation Aware Bayesian Affinity Estimation

so as to share the recommendation to others. Additionally, if the movie was strongly disappointing, the user is also more likely to share the low rating so as to warn others about it. On the other hand, if the movie does not evoke a strong reaction from the user, the likelihood that he or she will rate the movie at all is significantly lower. This behavior suggests that the event of observing a rating depends on the value of the rating. In this section, we extend the SABAE framework to incorporate this self-selecting property by explicitly modeling the probability of observing an affinity between a pair of entities. The resulting observation aware Bayesian affinity estimation (OA-BAE) framework relaxes the MAR assumption, thereby improving the prediction accuracy of the missing affinities.

We begin by extending the latent Dirichlet attribute aware Bayesian affinity estimation framework introduced in section 3.2 to account for the probability of observing a particular affinity. Without loss of generality, we assume that the affinities are recorded by the entities in the set $\mathcal{E}_1$ for the entities in the set $\mathcal{E}_2$. For the movie recommendation example, the set $\mathcal{E}_1$ corresponds to the users while the set $\mathcal{E}_2$ represents the movies. To model the probability of observing an affinity y_{mn}, let the associated weight w_{mn} be a Bernoulli random variable such that it takes a value one in the event the affinity is observed and is zero for a missing affinity.

Since, the probability of observing the affinity is expected to be high for both strongly positive and negative propensities than for a neutral affinity, an inverted Gaussian function of the affinity value can be used to efficiently model such a dependence. However, the sense of *neutrality* is often a property specific to entities recording the affinities (for example, in a movie recommendation engine, some users are heavy recorders characterized by large number of recorded ratings including for movies for which they have neutral affinities). Such a behavior can be easily captured by learning the expected value of the inverted Gaussian function which represents the value of a neutral affinity. On the other hand, an entity soliciting the affinities also influences the observation probability by evoking specific reactions in entities recording the affinities. Continuing with the movie recommendation example, some movies evoke a strong reaction in users resulting in a large number of recorded ratings. This suggests that a slight deviation from the neutral affinity value for such entities results in a high observation probability. This property can be learnt by modeling the spread of the Gaussian function using the variance term.

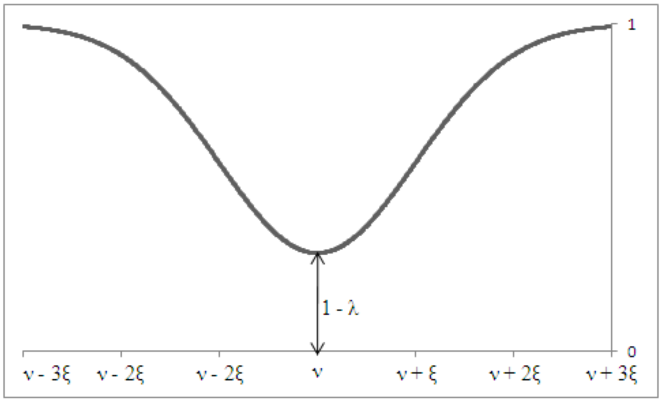

Figure 6: Probability of observing an affinity as a function of the affinity value.

This term controls the degree of monotonicity of the tails of the Gaussian curve as the affinities move away from the lowest neutral point.

The probability of observing the affinity y_{mn} is then modeled by the following parameterized form (see figure 6):

$$p(w_{mn}|y_{mn}, z_{1m}, z_{2n}) = 1 - \lambda_{z_{1m}z_{2n}} \exp\left(\frac{-(y_{mn} - \nu_{z_{1m}})^2}{2\xi^2_{z_{2n}}}\right) \tag{62}$$

where, $\nu_{z_{1m}}$ is the expected value of the inverted Gaussian function and is shared for the entities recording the affinities and assigned to the same cluster. Similarly, $\xi_{z_{2n}}$ is the variance of the function and is shared by the entities soliciting the affinities and assigned to the cluster. Finally, the strength of the Gaussian function is captured by the parameter $\lambda_{kl} \in [0, 1]$ for the affinities assigned to the co-cluster defined by the cluster assignments of the entity pairs. Incorporating the observation probability model into the LD-AA-BAE framework, the resulting graphical model is shown in figure 5. The complete likelihood for all observed and latent variables is then given by:

$$p(\mathcal{Y}, \mathcal{W}, \mathcal{X}_1, \mathcal{X}_2, \mathcal{Z}_1, \mathcal{Z}_2, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2|\boldsymbol{\alpha}_1, \alpha_2, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\beta}, \boldsymbol{\nu}, \boldsymbol{\xi}, \boldsymbol{\lambda}) = \tag{63}$$

$$p(\boldsymbol{\pi}_1|\boldsymbol{\alpha}_1)p(\boldsymbol{\pi}_2|\boldsymbol{\alpha}_2)\left(\prod_m p(z_{1m}|\boldsymbol{\pi}_1)p_{\psi_1}(\boldsymbol{x}_{1m}|\boldsymbol{\theta}_{1z_{1m}})\right)\left(\prod_n p(z_{2n}|\boldsymbol{\pi}_2)p_{\psi_2}(\boldsymbol{x}_{2n}|\boldsymbol{\theta}_{2z_{2n}})\right)$$

$$\left(\prod_{m,n} p_{\psi_{\mathcal{Y}}}(y_{mn}|\beta^{\dagger}_{z_{1m}z_{2n}}\boldsymbol{x}_{mn})p(w_{mn}|y_{mn}, \nu_{z_{1m}}, \xi_{z_{2n}}, \lambda_{z_{1m}z_{2n}})\right)$$

6.1 Inference and Learning

Maximization of the observed log-likelihood yields free model parameters. To overcome the intractability of this direct optimization, tractable lower bounds are obtained using a mean field approximation to the true posterior distribution over the latent variables. The posterior distribution is then approximated using a fully factorized distribution having the following form:

$$q(\mathcal{Y}_{\text{unobs}}, \mathcal{Z}_1, \mathcal{Z}_2, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2) = q^*(\boldsymbol{\pi}_1|\gamma_1)q^*(\boldsymbol{\pi}_{2|}\gamma_2)\left(\prod_{\substack{m,n \\ y_{mn}\in\mathcal{Y}_{\text{unobs}}}} q^*(y_{mn})\right)\left(\prod_m q^*(z_{1m}|r_{1m})\right)\left(\prod_n q^*(z_{2n}|r_{2n})\right)$$

where similar to the case of LD-AA-BAE, $q^*(\boldsymbol{\pi}_1|\gamma_1)$ and $q^*(\boldsymbol{\pi}_{2|}\gamma_2)$ are Dirichlet distributions with variational parameters γ_{1k} and γ_{2l} respectively while the variational distributions corresponding to the tightest lower bound, for the cluster assignment variables are discrete distributions with parameters $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ respectively. The optimal variational distribution for the unobserved affinities is given by:

$$q^*(y_{mn}) \propto p_0(y_{mn}) \exp\left(\langle y_{mn}, \sum_{k,l=1}^{K,L} r_{1mk} r_{2nl} (\boldsymbol{\beta}_{kl}^{\dagger} \boldsymbol{x}_{mn}) \rangle - \sum_{k,l=1}^{K,L} r_{1mk} r_{2nl} \frac{(y_{mn} - \nu_k)^2}{2\xi_l^2} \right) \tag{64}$$

where $p_0(y_{mn})$ is the Radon-Nikodym derivative with respect to the reference measure for the exponential family distribution assumed over the affinities. If the distribution is assumed to be Gaussian, i.e. $p_{\psi_y}(y_{mn}|\boldsymbol{\beta}_{kl}^{\dagger}\boldsymbol{x}_{mn}) = N(y_{mn}|\boldsymbol{\beta}_{kl}^{\dagger}\boldsymbol{x}_{mn}, \sigma_{kl}^2)$, the optimal variational distribution $q^*(y_{mn})$ is also a Normal distribution with mean and variance given as follows:

$$q^*(y_{mn}) = N\left(y_{mn} \left| \frac{\sum_{k,l=1}^{K,L} r_{1mk} r_{2nl} \left(\xi_l^2 \boldsymbol{\beta}_{kl}^{\dagger} \boldsymbol{x}_{mn} + \sigma_{kl}^2 \nu_k \right)}{\sum_{k,l=1}^{K,L} r_{1mk} r_{2nl} \left(\xi_l^2 + \sigma_{kl}^2 \right)}, \frac{1}{\sum_{k,l=1}^{K,L} r_{1mk} r_{2nl} \left(\frac{1}{\xi_l^2} + \frac{1}{\sigma_{kl}^2} \right)} \right. \right)$$

Following analysis in 3.1, the mean field equations for the variational parameters $(\gamma_{1k}, \gamma_{2l}, r_{1mk}, r_{2nl})$ is obtained by following equations:

$$\gamma_{1k} = \alpha_{1k} + \sum_{m=1}^{M} r_{1mk} \tag{65}$$

$$\gamma_{2l} = \alpha_{2l} + \sum_{n=1}^{N} r_{2nl} \tag{66}$$

$$\begin{aligned} r_{1mk} \propto \exp\Bigg(& \log p_{\psi_1}(\boldsymbol{x}_{1m}|\boldsymbol{\theta}_{1k}) + \gamma_{1k} + \sum_{n=1}^{N} \sum_{l=1}^{L} r_{2nl} \Bigg\{ w_{mn} \left(\log p_{\psi_y}(y_{mn}|\boldsymbol{\beta}_{kl}^{\dagger}\boldsymbol{x}_{mn}) + \log\left(1 - \lambda_{kl} \exp\left(\frac{-(y_{mn} - \nu_k)^2}{2\xi_l^2} \right) \right) \right) \\ & + (1 - w_{mn}) \mathbb{E}_q \left[\log p_{\psi_y}(y_{mn}|\boldsymbol{\beta}_{kl}^{\dagger}\boldsymbol{x}_{mn}) + \log\left(\lambda_{kl} \exp\left(\frac{-(y_{mn} - \nu_k)^2}{2\xi_l^2} \right) \right) \right] \Bigg\} \Bigg) \end{aligned} \tag{67}$$

$$\begin{aligned} r_{2nl} \propto \exp\Bigg(& \log p_{\psi_2}(\boldsymbol{x}_{2n}|\boldsymbol{\theta}_{2l}) + \gamma_{2l} + \sum_{m=1}^{M} \sum_{k=1}^{K} r_{1mk} \Bigg\{ w_{mn} \left(\log p_{\psi_y}(y_{mn}|\boldsymbol{\beta}_{kl}^{\dagger}\boldsymbol{x}_{mn}) + \log\left(1 - \lambda_{kl} \exp\left(\frac{-(y_{mn} - \nu_k)^2}{2\xi_l^2} \right) \right) \right) \\ & + (1 - w_{mn}) \mathbb{E}_q \left[\log p_{\psi_y}(y_{mn}|\boldsymbol{\beta}_{kl}^{\dagger}\boldsymbol{x}_{mn}) + \log\left(\lambda_{kl} \exp\left(\frac{-(y_{mn} - \nu_k)^2}{2\xi_l^2} \right) \right) \right] \Bigg\} \Bigg) \end{aligned} \tag{68}$$

Iteratively satisfying the mean field equations, one can attain a lower bound on the observed log-likelihood. This lower bound can then be used in place of the actual likelihood for parameter estimation. Maximizing the lower bound with respect to the model parameters results in following update equations:

$$\boldsymbol{\theta}_{1k} = \nabla\psi_1^{-1}\left(\frac{\sum_{m=1}^{M} r_{1mk} \boldsymbol{x}_{1m}}{\sum_{m=1}^{M} r_{1mk}} \right) \tag{69}$$

$$\boldsymbol{\theta}_{2l} = \nabla\psi_2^{-1}\left(\frac{\sum_{n=1}^{N} r_{2nl} \boldsymbol{x}_{2n}}{\sum_{n=1}^{N} r_{2nl}} \right) \tag{70}$$

Algorithm 5 Learn OA-BAE

```
Input: $\mathcal{Y}_{\text{obs}}, \mathcal{X}_1, \mathcal{X}_2, K, L$
Output: $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2, \boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\nu}, \boldsymbol{\xi}$
        $[m]_1^M, [n]_1^N, [k]_1^K, [l]_1^L$

  Step 0: Initialize $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2, \boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\nu}, \boldsymbol{\xi}$
  Until Convergence
      Step 1: E-Step
          Step 1a: Initialize $r_{1mk}, r_{2nl}$
          Until Convergence
              Step 1b: Update $q^*(y_{mn})$ using equation (64)
              Step 1c: Update $(\gamma_{1k}, \gamma_{2l})$ using equations (65) and (66)
              Step 1d: Update $(r_{1mk}, r_{2nl})$ using equations (67) and (68)
      Step 2: M-Step
          Step 2a: Update $(\boldsymbol{\theta}_{1k}, \boldsymbol{\theta}_{2l})$ using equations (69) and (70)
          Step 2b: Update $\boldsymbol{\beta}_{kl}$ using equation (71)
          Step 2c: Update $(\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2)$ using equations (72) and (73)
          Step 2d: Update $\lambda_{kl}$ using equation (74)
          Step 2e: Update $\nu_k$ using equation (75)
          Step 2f: Update $\xi_l$ using equation (76)
```

$$\boldsymbol{\beta}_{kl} = \operatorname*{arg\,max}_{\beta \in \mathbb{R}^D} \sum_{m=1}^{M} \sum_{n=1}^{N} r_{1mk} r_{2nl} \left[\left\langle \left(w_{mn} y_{mn} + (1 - w_{mn}) \mathbb{E}_q[y_{mn}] \right), \boldsymbol{\beta}^\dagger \boldsymbol{x}_{mn} \right\rangle - \psi_{\mathcal{Y}} \left(\boldsymbol{\beta}^\dagger \boldsymbol{x}_{mn} \right) \right] \tag{71}$$

$$\boldsymbol{\alpha}_1 = \operatorname*{arg\,max}_{\boldsymbol{\alpha}_1 \in \mathbb{R}_{++}^K} \left(\log \frac{\Gamma(\sum_{k=1}^K \alpha_{1k})}{\prod_{k=1}^K \Gamma(\alpha_{1k})} + \sum_{k=1}^{K} \left(\alpha_{1k} + \sum_{m=1}^{M} r_{1mk} - 1 \right) \left(\Psi(\gamma_{1k}) - \Psi \left(\sum_{k'=1}^{K} \gamma_{1k'} \right) \right) \right) \tag{72}$$

$$\boldsymbol{\alpha}_2 = \operatorname*{arg\,max}_{\boldsymbol{\alpha}_2 \in \mathbb{R}_{++}^L} \left(\log \frac{\Gamma(\sum_{l=1}^L \alpha_{2l})}{\prod_{l=1}^L \Gamma(\alpha_{2l})} + \sum_{l=1}^{L} \left(\alpha_{2l} + \sum_{n=1}^{N} r_{2nl} - 1 \right) \left(\Psi(\gamma_{2l}) - \Psi \left(\sum_{l'=1}^{L} \gamma_{2l'} \right) \right) \right) \tag{73}$$

A closed form expression cannot be obtained for the updates of the parameters associated with observability model $(\lambda_{kl}, \nu_k, \xi_l)$. However, the following constrained optimization problems can be solved efficiently using the Newton-Raphson's method.

$$\lambda_{kl} = \operatorname*{arg\,max}_{\lambda_{kl} \in [0,1]} \sum_{m=1}^{M} \sum_{n=1}^{N} r_{1mk} r_{2nl} \left(w_{mn} \log \left(1 - \lambda_{kl} \exp \left(\frac{-(y_{mn} - \nu_k)^2}{2\xi_l^2} \right) \right) + (1 - w_{mn}) \log \lambda_{kl} \right) \tag{74}$$

$$\nu_k = \operatorname*{arg\,max}_{\nu_k \in \mathbb{R}} \sum_{m=1}^{M} \sum_{n=1}^{N} \sum_{l=1}^{L} r_{1mk} r_{2nl} \left(w_{mn} \log \left(1 - \lambda_{kl} \exp \left(\frac{-(y_{mn} - \nu_k)^2}{2\xi_l^2} \right) \right) + \frac{(1 - w_{mn})}{2\xi_l^2} \mathbb{E}_q \left[-(y_{mn} - \nu_k)^2 \right] \right) \tag{75}$$

$$\xi_l = \operatorname*{arg\,max}_{\xi_l \in \mathbb{R}} \sum_{m=1}^{M} \sum_{n=1}^{N} \sum_{k=1}^{K} r_{1mk} r_{2nl} \left(w_{mn} \log \left(1 - \lambda_{kl} \exp \left(\frac{-(y_{mn} - \nu_k)^2}{2\xi_l^2} \right) \right) + \frac{(1 - w_{mn})}{2\xi_l^2} \mathbb{E}_q \left[-(y_{mn} - \nu_k)^2 \right] \right) \tag{76}$$

The resulting EM algorithm for learning the model parameters is given in algorithm 5.

7 Bayesian Affinity Estimation with Temporal Dynamics [‡]

Affinity relationships between sets of entities are dynamic in nature with constantly evolving preferences. For example, a multitude of datasets recording user preferences for items indicate a strong temporal behavior [6]. Popularity of different items is constantly changing as new selections emerge, in turn resulting in a change in the user preferences. Static affinity estimation frameworks ignore this dynamic nature of the data and hence suffer from inferior predictive capability. Thus, modeling temporal dynamics is an important step towards accurate affinity relationship modeling. The importance of modeling the dynamic behavior for affinity estimation is evident from the recently concluded Netflix challenge, where temporal modeling had a significant role in the grand-prize winning solution [33]. This section extends the SABAE framework for modeling the dynamic behavior of affinity expressing datasets. Specifically, temporal dynamics are incorporated into the LD-AA-BAE framework to specify a statistical model of *cluster evolution*.

To model the temporal dynamics, it is assumed that the data is divided into different time slices. For example, in a user-item system, the different time slices might correspond to different months of the year to account for the seasonal effects in the user preferences or item popularity. Two changes are made to the LD-AA-BAE model to account for the temporal behavior. First, the Dirichlet distribution priors over the mixing coefficients $\boldsymbol{\pi}_1$ and $\boldsymbol{\pi}_2$ are replaced by logistic-normal priors [34] with mean parameters $\boldsymbol{\alpha}_1$ and $\boldsymbol{\alpha}_2$ respectively, and secondly, within each time slice t, the affinity relationships are modeled using a static LD-AA-BAE model with a logistic-normal prior, where the entity clusters associated with the time slice t evolve form clusters associated with slice $t-1$. Cluster evoluation is encoded by assuming a linear dynamic model over the mean parameters $\boldsymbol{\alpha}_1$ and $\boldsymbol{\alpha}_2$ of the logistic-normal priors along with an evolution of the co-cluster GLM coefficients $\boldsymbol{\beta}$. The dynamics associated with the model are then given by

$$\begin{aligned}
\boldsymbol{\alpha}_{1,t}|\boldsymbol{\alpha}_{1,t-1} &\sim N(\boldsymbol{\alpha}_{1,t}|\boldsymbol{\alpha}_{1,t-1}, \delta_1^2 I) && (77)\\
\boldsymbol{\alpha}_{2,t}|\boldsymbol{\alpha}_{2,t-1} &\sim N(\boldsymbol{\alpha}_{2,t}|\boldsymbol{\alpha}_{2,t-1}, \delta_2^2 I) && (78)\\
\boldsymbol{\beta}_{kl,t}|\boldsymbol{\beta}_{kl,t-1} &\sim N(\boldsymbol{\beta}_{kl,t}|\boldsymbol{\beta}_{kl,t-1}, \omega^2 I) && (79)
\end{aligned}$$

The variances in the dynamic model can be set using cross-validation and are then held fixed. The attribute parameters $\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2$, however are assumed to be static, since entity attributes (such as user zip code and movie release year or running time) are not expected to evolve over time. The graphical model for a dynamic Bayesian affinity estimation model is shown in figure 7. When the horizontal arrows representing time evolution are removed, the graphical model reduces to a set of independent attribute aware Bayesian affinity estimation models with a logistic-normal priors. Modeling temporal dynamics, causes the clusters to evolve smoothly over time.

7.1 Approximate Inference using Variational Kalman Filtering

The non-conjugacy of the logistic-normal prior over the mixing coefficients and the multinomial distribution for the cluster assignments renders exact posterior inference intractable. Non-conjugacy further complicates the use of stochastic sampling methods based on Gibbs sampling. To overcome this problem, we formulate a parameterized variational approximation to the true posterior distribution. The variational parameters are then estimated to minimize the KL divergence between the true posterior distribution over the latent variables and the assumed variational distribution. In the dynamic Bayesian

[‡] Joint work with Yubin Park, yubin.park@gmail.com

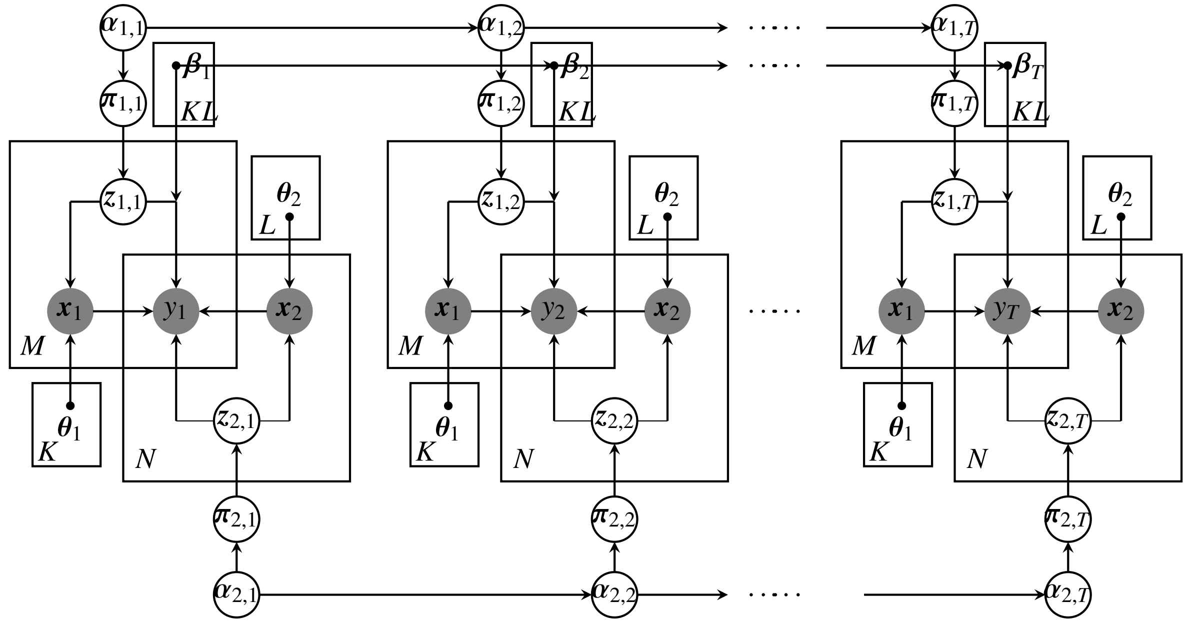

Figure 7: Graphical model for Dynamic Bayesian Affinity Estimation

affinity estimation model, the latent variables are the logistic-normal prior mean $\boldsymbol{\alpha}_{1,1:T}, \boldsymbol{\alpha}_{2,1:T}$, the mixing coefficients $\boldsymbol{\pi}_{1,1:T}, \boldsymbol{\pi}_{2,1:T}$, the cluster assignments $\mathcal{Z}_{1,1:T}, \mathcal{Z}_{2,1:T}$ and the GLM coefficients $\boldsymbol{\beta}_{kl,1:T}$. The approximate variation posterior distribution is

$$q(\boldsymbol{\alpha}_{1,1:T}, \boldsymbol{\alpha}_{2,1:T}, \mathcal{Y}^{1:T}_{\text{unobs}}, \boldsymbol{\pi}_{1,1:T}, \boldsymbol{\pi}_{2,1:T}, \mathcal{Z}_{1,1:T}, \mathcal{Z}_{2,1:T}, \boldsymbol{\beta}_{kl,1:T}) = \tag{80}$$

$$q(\boldsymbol{\alpha}_{1,1} \dots \boldsymbol{\alpha}_{1,T} | \hat{\boldsymbol{\alpha}}_{1,1} \dots \hat{\boldsymbol{\alpha}}_{1,T}) \times q(\boldsymbol{\alpha}_{2,1} \dots \boldsymbol{\alpha}_{2,T} | \hat{\boldsymbol{\alpha}}_{2,1} \dots \hat{\boldsymbol{\alpha}}_{2,T}) \times \left(\prod_{t=1}^{T} \prod_{\substack{m,n \\ y_{mn,t} \in \mathcal{Y}^{t}_{\text{unobs}}}} q(y_{mn,t} | \vartheta_{mn,t}, \varsigma_{mn,t}) \right) \times$$

$$\prod_{t=1}^{T} \left(q(\boldsymbol{\pi}_{1,t} | \boldsymbol{\rho}_{1,t}, \boldsymbol{\Lambda}_{1,t}) q(\boldsymbol{\pi}_{2,t} | \boldsymbol{\rho}_{2,t}, \boldsymbol{\Lambda}_{2,t}) \left(\prod_{m=1}^{M} q(z_{1m,t} | r_{1mk,t}) \right) \left(\prod_{n=1}^{N} q(z_{2n,t} | r_{2nl,t}) \right) \right) \times \prod_{k=1}^{K} \prod_{l=1}^{L} q(\boldsymbol{\beta}_{kl,1} \dots \boldsymbol{\beta}_{kl,T} | \hat{\boldsymbol{\beta}}_{kl,1} \dots \hat{\boldsymbol{\beta}}_{kl,T})$$

where, the variational distribution over the missing affinities is a Gaussian with mean ϑ_{mn} and variance ς_{mn}, $q(\boldsymbol{\pi}_{1,t} | \boldsymbol{\rho}_{1,t}, \boldsymbol{\Lambda}_{1,t}) q(\boldsymbol{\pi}_{2,t} | \boldsymbol{\rho}_{2,t}, \boldsymbol{\Lambda}_{2,t})$ are multivariate Gaussian distributions with means $\boldsymbol{\rho}_{1,t}, \boldsymbol{\rho}_{2,t}$ and diagonal covariance matrices $\boldsymbol{\Lambda}_{1,t}, \boldsymbol{\Lambda}_{2,t}$ respectively. The variational distributions for cluster assignment variables are discrete distributions with parameters $r_{1mk,t}$ and $r_{2nl,t}$ respectively. The dynamics are captured by "Gaussian variational observations" $\hat{\boldsymbol{\alpha}}_{1,1:T}, \hat{\boldsymbol{\alpha}}_{2,1:T}, \hat{\boldsymbol{\beta}}_{kl,1:T}$. The *variational observations* $\hat{\alpha}_{1,1:T}$, $\hat{\alpha}_{2,1:T}$ and $\hat{\beta}_{kl,1:T}$ can be expressed by following Gaussian distributions:

$$\hat{\alpha}_{1,t} | \alpha_{1,t} \sim N(\hat{\alpha}_{1,t} | \alpha_{1,t}, \hat{v}^2_{\alpha_{1},t} I) \tag{81}$$

$$\hat{\alpha}_{2,t} | \alpha_{2,t} \sim N(\hat{\alpha}_{2,t} | \alpha_{2,t}, \hat{v}^2_{\alpha_{2},t} I) \tag{82}$$

$$\hat{\beta}_{kl,t} | \beta_{kl,t} \sim N(\hat{\beta}_{kl,t} | \beta_{kl,t}, \hat{v}^2_{\beta_{kl},t} I) \tag{83}$$

where $\hat{v}^2_{\alpha_1,t}$, $\hat{v}^2_{\alpha_2,t}$ and $\hat{v}^2_{\beta_{kl},t}$ are the variances of the three distributions respectively and can be seen as the variational parameters associated with these distributions. These variational observations and parameters are not observed basically. As in Blei and Lafferty [9], these values are "hypothetical outputs" to facilitate inferencing using a linear state space model such as Kalman filtering [41].

These hypothetical outputs correspond to a variational Gaussian distribution over the dynamic latent variables. This choice of variational distribution allows a direct application of Kalman filtering to update the linear state space model expressed by these variational distributions. Then, the lower bound can be expressed in terms of the variational parameters (means and variances of these Gaussian distributions) which can be defined by following expressions:

$$\tilde{m}_{\alpha_{.},t} \equiv \mathbb{E}(\alpha_{.,t}|\hat{\alpha}_{.,1:T}) \tag{84}$$

$$\tilde{V}_{\alpha_{.},t} \equiv \mathbb{E}((\alpha_{.,t} - \tilde{m}_{\alpha_{.},t})^2|\hat{\alpha}_{.,1:T}) \tag{85}$$

$$\tilde{m}_{\beta_{kl},t} \equiv \mathbb{E}(\beta_{kl,t}|\hat{\beta}_{kl,1:T}) \tag{86}$$

$$\tilde{V}_{\beta_{kl},t} \equiv \mathbb{E}((\beta_{kl,t} - \tilde{m}_{\beta_{kl},t})^2|\hat{\beta}_{kl,1:T}). \tag{87}$$

The Gaussian assumption over the dynamic latent variables has three positive aspects. First, by using such a linear model, we can readily compute $\tilde{m}_{\alpha_{.},t}$, $\tilde{V}_{\alpha_{.},t}$, $\tilde{m}_{\beta_{kl},t}$ and $\tilde{V}_{\beta_{kl},t}$ by the standard Kalman filter equations. Kalman filter equations give us the recursions for $\tilde{m}_{\alpha_{.},t}$, $\tilde{V}_{\alpha_{.},t}$, $\tilde{m}_{\beta_{kl},t}$ and $\tilde{V}_{\beta_{kl},t}$, which are functions of $\hat{\alpha}_{1,1:T}$, $\hat{\alpha}_{2,1:T}$, $\hat{\beta}_{kl,1:T}$, $\hat{v}^2_{\alpha_1,t}$, $\hat{v}^2_{\alpha_2,t}$ and $\hat{v}^2_{\beta_{kl},t}$. Second, using the symmetry properties of the Gaussian density, $f_{\mu,\Sigma}(x) = f_{x,\Sigma}(\mu)$, we can easily maximize the lower bound on the observed log-likelihood with respect to variational Gaussian observations and parameters. Finally, the additional variational parameters ($\hat{v}^2_{\alpha_1,t}$, $\hat{v}^2_{\alpha_2,t}$, $\hat{v}^2_{\beta_{kl},t}$), provide an additional degree of freedom to optimize over, resulting in a tighter lower bound.

Therefore, the $\tilde{m}_{\alpha_{.}}$, $\tilde{V}_{\alpha_{.}}$, $\tilde{m}_{\beta_{kl}}$, $\tilde{V}_{\beta_{kl}}$ can be obtained by maximizing the lower bound resulting in the following coupled equations (For a detailed derivation of the Kalman filtering updates and the corresponding lower bound please refer to Appendix E):

$$\frac{\partial \mathcal{L}(\tilde{m}_{\alpha_{.}}, \tilde{V}_{\alpha_{.}})}{\partial \hat{\alpha}_{.}} = 0 \text{ and } \quad \frac{\partial \mathcal{L}(\tilde{m}_{\alpha_{.}}, \tilde{V}_{\alpha_{.}})}{\partial \hat{v}_{\alpha_{.}}} = 0 \tag{88}$$

$$\frac{\partial \mathcal{L}(\tilde{m}_{\beta}, \tilde{V}_{\beta})}{\partial \hat{\beta}} = 0 \quad \text{and} \quad \frac{\partial \mathcal{L}(\tilde{m}_{\beta}, \tilde{V}_{\beta})}{\partial \hat{v}_{\beta}} = 0 \tag{89}$$

Assuming a fixed value of the free model parameters, we next derive an optimal lower bound corresponding to the factorized approximation to the true posterior distribution over the latent variables. The optimal lower bound can then be optimized over the free model parameters to learn the model parameters.

We next move on to the updates for the remaining variational parameters to obtain an optimal lower bound to the observed log-likelihood. Since the mixing coefficients are sampled from a logistic-normal prior, the log-likelihood of the cluster assignment variables $\mathcal{Z}_1$, is obtained as follows:

$$\log p(\mathcal{Z}_{1,1:T}|\boldsymbol{\pi}_{1,1:T}) = \sum_{t=1}^{T}\sum_{m=1}^{M}\sum_{k=1}^{K} z_{1mk,t}\left(\pi_{1k,t} - \log\left(\sum_{k'=1}^{K}\exp(\pi_{1k',t})\right)\right)$$

Since, log is a concave function, using Jensen's inequality [30], a family of lower bounds can be obtained for the above log-likelihood expression:

$$\log p(\mathcal{Z}_{1,1:T}|\boldsymbol{\pi}_{1,1:T}) \geq \sum_{t=1}^{T}\sum_{m=1}^{M}\sum_{k=1}^{K} z_{1mk,t}\left(\pi_{1k,t} - \varepsilon_{1,t}^{-1}\sum_{k'=1}^{K}\exp(\pi_{1k,t}) + 1 - \log\varepsilon_{1,t}\right)$$

where $\varepsilon_{1,t}$, $[t]_1^T$ are the variational parameters corresponding to the family of lower bounds. The variational parameters corresponding to the tightest lower bound is then given by:

$$\varepsilon_{1,t} \quad = \quad \sum_{k=1}^{K}\exp\left(\rho_{1k,t} + \frac{\Lambda_{1k,t}}{2}\right) \tag{90}$$

$$\varepsilon_{2,t} \quad = \quad \sum_{l=1}^{L}\exp\left(\rho_{2l,t} + \frac{\Lambda_{2l,t}}{2}\right) \tag{91}$$

where $\varepsilon_{1,t}$, $[t]_1^T$ are the variational parameters corresponding to the cluster assignment variables $\mathcal{Z}_2$. Proceeding with the maximization of the lower bound on the observed log-likelihood with respect to the variational parameters corresponding to missing affinities, the following updates for the variational mean and variance can then be obtained:

$$\vartheta_{mn,t} = \frac{\sum_{k,l=1}^{K,L}\frac{r_{1mk,t}r_{2nl,t}}{\sigma_{kl}^2}\left(\tilde{\boldsymbol{m}}_{\boldsymbol{\beta}_{kl},t}^{\dagger}\boldsymbol{x}_{mn}\right)}{\sum_{k,l=1}^{K,L}\frac{r_{1mk,t}r_{2nl,t}}{\sigma_{kl}^2}} \tag{92}$$

$$\varsigma_{mn,t}^2 = \frac{1}{\sum_{k,l=1}^{K,L}\frac{r_{1mk,t}r_{2nl,t}}{\sigma_{kl}^2}} \tag{93}$$

Using the variational Gaussian distribution of the missing affinities, the following expression evaluates the expectation of the log-likelihood function of the affinites.

$$\mathbb{E}_q[(y_{mn,t} - \boldsymbol{\beta}_{kl,t}^{\dagger}\boldsymbol{x}_{mn})^2] = w_{mn}y_{mn,t}^2 + (1 - w_{mn})(\varsigma_{mn,t}^2 + \vartheta_{mn,t}^2) \tag{94}$$

$$-2(w_{mn}y_{mn,t} + (1 - w_{mn})\vartheta_{mn,t})(\tilde{\boldsymbol{m}}_{\boldsymbol{\beta}_{kl},t}^{\dagger}\boldsymbol{x}_{mn}) + \boldsymbol{x}_{mn}^{\dagger}(\tilde{\boldsymbol{V}}_{\boldsymbol{\beta}_{kl},t} + \tilde{\boldsymbol{m}}_{\boldsymbol{\beta}_{kl},t}\tilde{\boldsymbol{m}}_{\boldsymbol{\beta}_{kl},t}^{\dagger})\boldsymbol{x}_{mn}$$

Note that the expression uses the known affinity values for non-missing affinities (represented by $w_{mn} = 0$) and for the missing affinities, relevant expectations are taken under the variational distribution. Maximization with respect to the variational discrete distributions over the cluster assignment variables yields following updates for the variational parameters $\boldsymbol{r}_{1m}$ and $\boldsymbol{r}_{2n}$:

$$r_{1mk,t} \propto \exp\left(\rho_{1k,t} + \log p_{\psi_1}(\boldsymbol{x}_{1m}|\boldsymbol{\theta}_{1k}) + \sum_{n=1}^{N}\sum_{l=1}^{L} r_{2nl,t}\left(-\frac{1}{2}\log\sigma_{kl}^2 - \frac{1}{2\sigma_{kl^2}}\mathbb{E}_q[(y_{mn,t} - \boldsymbol{\beta}_{kl,t}^{\dagger}\boldsymbol{x}_{mn})^2]\right)\right) \tag{95}$$

$$r_{2nl,t} \propto \exp\left(\rho_{2l,t} + \log p_{\psi_2}(\boldsymbol{x}_{2n}|\boldsymbol{\theta}_{2l}) + \sum_{m=1}^{M}\sum_{k=1}^{K} r_{1mk,t}\left(-\frac{1}{2}\log\sigma_{kl}^2 - \frac{1}{2\sigma_{kl^2}}\mathbb{E}_q[(y_{mn,t} - \boldsymbol{\beta}_{kl,t}^{\dagger}\boldsymbol{x}_{mn})^2]\right)\right) \tag{96}$$

Similarly, the mean field equations for the mean parameters of the variational distribution over the mixing coefficients $\boldsymbol{\pi}_1$ and $\boldsymbol{\pi}_2$ corresponding to an optimal lower bound on the observed log-likelihood is given by following equations:

$$\rho_{1k,t} = \tilde{m}_{\alpha_{1k},t} + \sigma_1^2\sum_{m=1}^{M} r_{1mk,t} - W\left(-M\varepsilon_{1,t}\sigma_1^2\exp\left(\frac{\Lambda_{1k,t}}{2} + \tilde{m}_{\alpha_{1k},t} + \sigma_1^2\sum_{m=1}^{M} r_{1mk,t}\right)\right) \tag{97}$$

$$\rho_{2l,t} = \tilde{m}_{\alpha_{2l},t} + \sigma_2^2 \sum_{n=1}^{N} r_{2nl,t} - W\left(-N\varepsilon_{2,t}\sigma_2^2 \exp\left(\frac{\Lambda_{2l,t}}{2} + \tilde{m}_{\alpha_{2l},t} + \sigma_2^2 \sum_{n=1}^{N} r_{2nl,t}\right)\right) \tag{98}$$

where W is the Lambert's W function. The updates for the individual diagonal entries of the (diagonal) covariance matrices is obtained as a solution to following constrained optimization:

$$\Lambda_{1k,t} = \underset{\Lambda_{1k,t}\in\mathbb{R}_{++}}{\arg\max} -\frac{1}{2\sigma_1^2}\sum_{k=1}^{K}\Lambda_{1k,t} - M\varepsilon_{1,t}\sum_{k=1}^{K}\exp\left(\rho_{1k,t} + \frac{\Lambda_{1k,t}}{2}\right) + \frac{1}{2}\sum_{k=1}^{K}\log\Lambda_{1k,t} \tag{99}$$

$$\Lambda_{2l,t} = \underset{\Lambda_{2l,t}\in\mathbb{R}_{++}}{\arg\max} -\frac{1}{2\sigma_2^2}\sum_{l=1}^{L}\Lambda_{2l,t} - N\varepsilon_{2,t}\sum_{l=1}^{L}\exp\left(\rho_{2l,t} + \frac{\Lambda_{2l,t}}{2}\right) + \frac{1}{2}\sum_{l=1}^{L}\log\Lambda_{2l,t} \tag{100}$$

The set of coupled update equations for the variational parameters can be satisfied iteratively which yields a tight lower bound on the observed log-likelihood. Note that the convergence in guaranteed since the bound is convex with respect to the variational parameters [31]. The optimal bound can be maximized to get an improved estimate of the model parameters. The following updates are obtained for the natural parameters of the exponential family distributions over the entity attributes:

$$\boldsymbol{\theta}_{1k} = \nabla\psi_1^{-1}\left(\frac{\sum_{t=1}^{T}\sum_{m=1}^{M} r_{1mk,t}\boldsymbol{x}_{1m}}{\sum_{t=1}^{T}\sum_{m=1}^{M} r_{1mk,t}}\right) \tag{101}$$

$$\boldsymbol{\theta}_{2l} = \nabla\psi_2^{-1}\left(\frac{\sum_{t=1}^{T}\sum_{n=1}^{N} r_{2nl,t}\boldsymbol{x}_{2n}}{\sum_{t=1}^{T}\sum_{n=1}^{N} r_{2nl,t}}\right) \tag{102}$$

The variances associated with the logistic-normal prior and the affinities within each co-cluster are updated by following equations respectively,

$$\sigma_1^2 = \frac{1}{KT}\left[\sum_{t=1}^{T}\left(\| \boldsymbol{\rho}_{1,t} - \tilde{\boldsymbol{m}}_{\alpha_1,t} \|^2 + 2\mathrm{Tr}(\tilde{\boldsymbol{V}}_{\alpha_1,t}) + \mathrm{Tr}(\boldsymbol{\Lambda}_{1,t})\right)\right] \tag{103}$$

$$\sigma_2^2 = \frac{1}{LT}\left[\sum_{t=1}^{T}\left(\| \boldsymbol{\rho}_{2,t} - \tilde{\boldsymbol{m}}_{\alpha_2,t} \|^2 + 2\mathrm{Tr}(\tilde{\boldsymbol{V}}_{\alpha_2,t}) + \mathrm{Tr}(\boldsymbol{\Lambda}_{2,t})\right)\right] \tag{104}$$

$$\sigma_{kl}^2 = \frac{\sum_{t=1}^{T}\sum_{m,n=1}^{M,N} r_{1mk,t}r_{2nl,t}\left(\mathbb{E}_q[(y_{mn,t} - \boldsymbol{\beta}_{kl,t}^{\dagger}\boldsymbol{x}_{mn})^2]\right)}{\sum_{t=1}^{T}\sum_{m,n=1}^{M,N} r_{1mk,t}r_{2nl,t}} \tag{105}$$

8 Entity Attributes Estimation

In most datasets recording affinities between sets of entities, the auxiliary entity attributes are collected from different connected sources. Often such sources are noisy and incomplete resulting in a large number of missing entity attributes. For example, in internet-based applications such as online recommender systems, online advertisement targeting etc., the attributes associated with the targeted customer base is often collected from the user profiles. The users are asked to submit the profile information at the time of their registration with the system. In many cases, users can choose not to provide this information resulting in missing user attributes. Similarly, attribute information might be unavailable for many items or products due to un-documented attributes or as a result of a noisy data collection process. In such a scenario, it is imperative to impute the missing entities so as to aid in the prediction of the missing affinities.

This section extends the LD-AA-BAE framework to automatically impute the missing attributes, for both entity sets $\mathcal{E}_1$ and $\mathcal{E}_2$. We assume that for an entity with missing attributes, at least a single associated affinity is observed. This is a reasonable assumption, since a lack of attribute as well as any prior recorded affinity information means that the entity is absent from the system with no available data. Subsets $\mathcal{X}_{1\text{unobs}} \subseteq \mathcal{X}_1$ and $\mathcal{X}_{2\text{unobs}} \subseteq \mathcal{X}_2$ denote the missing attributes for two entity sets respectively. Similar to the weights associated with the affinities, we assign a weight $w_{1m}(w_{2n})$ with each entity $e_{1m} \in \mathcal{E}_1(e_{2n} \in \mathcal{E}_2)$ such that $w_{1m} = 0((w_{2n}) = 0)$ if $\boldsymbol{x}_{1m} \in \mathcal{X}_{1\text{unobs}}(\boldsymbol{x}_{2n} \in \mathcal{X}_{2\text{unobs}})$. For computational convinience during inference, we assume that the conditional distribution of the affinities conditioned on the cluster assignment variables is a Gaussian distribution.

$$p_{\psi_y}(y_{mn}|\boldsymbol{\beta}^{\dagger}_{z_{1m}z_{2n}}\boldsymbol{x}_{mn}) = N(y_{mn}|\boldsymbol{\beta}^{\dagger}_{z_{1m}z_{2n}}\boldsymbol{x}_{mn}, \sigma^2_{z_{1m}z_{2n}})$$

As before, the free model parameters can be learnt using a variational EM algorithm with a mean field approximation. Using a fully factorized variational distribution to approximate the true posterior distribution, a parameterized variational distribution is then defined over the latent variables.

$$q(\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \mathcal{Y}_{\text{unobs}}, \mathcal{X}_{1\text{unobs}}, \mathcal{X}_{2\text{unobs}}, \mathcal{Z}_1, \mathcal{Z}_2) = q^*(\boldsymbol{\pi}_{1m}|\boldsymbol{\gamma}_{1m})q^*(\boldsymbol{\pi}_{2n}|\boldsymbol{\gamma}_{2n})\times \tag{106}$$

$$\left(\prod_{\substack{m,n\\ y_{mn}\in\mathcal{Y}_{\text{unobs}}}} N(y_{mn}|\vartheta_{mn}, \varsigma^2_{mn})\right)\left(\prod_m q^*(z_{1m}|\boldsymbol{r}_{1m})\right)\left(\prod_n q^*(z_{2n}|\boldsymbol{r}_{2n})\right)\left(\prod_{\substack{m\\ \boldsymbol{x}_{1m}\in\mathcal{X}_{1\text{unobs}}}} q^*_{\psi^*_1}(\boldsymbol{x}_{1m}|\boldsymbol{\varpi}_{1m})\right)\left(\prod_{\substack{n\\ \boldsymbol{x}_{2n}\in\mathcal{X}_{2\text{unobs}}}} q^*_{\psi^*_2}(\boldsymbol{x}_{2n}|\boldsymbol{\varpi}_{2n})\right)$$

where $\vartheta_{mn}, \varsigma^2_{mn}$ are the mean and variance of the variational Gaussian distribution for the missing affinities while similar to the case of LD-AA-BAE, $q^*(\boldsymbol{\pi}_{1m}|\boldsymbol{\gamma}_{1m})$ and $q^*(\boldsymbol{\pi}_{2n}|\boldsymbol{\gamma}_{2n})$ are K and L dimensional Dirichlet distributions with parameters $\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2$ respectively. Likewise, $q^*(z_{1m}|\boldsymbol{r}_{1m}), q^*(z_{2n}|\boldsymbol{r}_{2n})$ are discrete distributions over the cluster assignments. The variational distributions over the missing entities are exponential family distributions with natural parameters ϖ_{1m} and ϖ_{2n} respectively. Optimization over the variational parameters yields a tight lower bound on the observed log-likelihood. The result is a set of coupled equations known as mean field equations that can be satisfied iteratively to yield the optimal lower bound.

For notational convenience, we use the notation $\mathbb{E}_q[\cdot]$ to denote the expectations of the functions of latent variables with respect to the variational distributions. The expectation yields the actual values for the observed variables and the difference will be clear from the context. Following inference methodology for LD-AA-BAE (section 3.2), the updates for the variational parameters is obtained as following equations.

$$\vartheta_{mn} = \frac{\sum_{k,l=1}^{K,L} \frac{r_{1mk}r_{2nl}}{\sigma^2_{kl}} \left(\boldsymbol{\beta}^{\dagger}_{1kl}\mathbb{E}_q[\boldsymbol{x}_{1m}] + \boldsymbol{\beta}^{\dagger}_{2kl}\mathbb{E}_q[\boldsymbol{x}_{2n}]\right)}{\sum_{k,l=1}^{K,L} \frac{r_{1mk}r_{2nl}}{\sigma^2_{kl}}} \tag{107}$$

$$\varsigma^2_{mn} = \frac{1}{\sum_{k,l=1}^{K,L} \frac{r_{1mk}r_{2nl}}{\sigma^2_{kl}}} \tag{108}$$

The variational distributions for the missing attributes that correspond to an optimal lower bound then assume the following forms of exponential family distributions:

$$q^*(\boldsymbol{x}_{1m}) \propto p_0(\boldsymbol{x}_{1m}) \exp\left(-\sum_{n=1}^{N}\sum_{k,l=1}^{K,L} \frac{r_{1mk}r_{2nl}}{\sigma^2_{kl}}(\boldsymbol{\beta}_{1kl}\boldsymbol{x}_{1m})^2\right) \exp\left(\langle \boldsymbol{x}_{1m}, \varpi_{1m}\rangle\right) \tag{109}$$

$$\varpi_{1m} = \sum_{k=1}^{K} r_{1mk} \left[\boldsymbol{\theta}_{1k} + \boldsymbol{\beta}_{1kl} \sum_{n=1}^{N} \sum_{l=1}^{L} \frac{r_{2nl}}{\sigma^2_{kl}} \left(\mathbb{E}_q[y_{mn}] - \boldsymbol{\beta}^{\dagger}_{2kl} \mathbb{E}_q[\boldsymbol{x}_{2n}] \right) \right] \tag{110}$$

$$q^*(\boldsymbol{x}_{2n}) \propto p_0(\boldsymbol{x}_{2n}) \exp\left(- \sum_{m=1}^{M} \sum_{k,l=1}^{K,L} \frac{r_{1mk} r_{2nl}}{\sigma^2_{kl}} (\boldsymbol{\beta}_{2kl} \boldsymbol{x}_{2n})^2 \right) \exp\left(\langle \boldsymbol{x}_{2n}, \varpi_{2n} \rangle \right) \tag{111}$$

$$\varpi_{2n} = \sum_{l=1}^{L} r_{2nl} \left[\boldsymbol{\theta}_{2l} + \boldsymbol{\beta}_{2kl} \sum_{m=1}^{M} \sum_{k=1}^{K} \frac{r_{1mk}}{\sigma^2_{kl}} \left(\mathbb{E}_q[y_{mn}] - \boldsymbol{\beta}^{\dagger}_{1kl} \mathbb{E}_q[\boldsymbol{x}_{1m}] \right) \right] \tag{112}$$

where ϖ_{1m} and ϖ_{2n} are the natural parameters of the distributions respectively and ψ^*_1, ψ^*_2 are the corresponding log partition functions. Finally, the updates for the parameters of the Dirichlet and discrete distributions are given by:

$$\gamma_{1k} = \alpha_{1k} + \sum_{m=1}^{M} r_{1mk} \tag{113}$$

$$\gamma_{2l} = \alpha_{2l} + \sum_{n=1}^{N} r_{2nl} \tag{114}$$

$$\begin{gathered} r_{1mk} \propto \exp\Big(\mathbb{E}_q[\log p_{\psi_1}(\boldsymbol{x}_{1m} | \boldsymbol{\theta}_{1k})] + \Psi(\gamma_{1k}) + \\ \sum_{n=1}^{N} \sum_{l=1}^{L} \frac{r_{2nl}}{\sigma^2_{kl}} \left(\mathbb{E}_q \left[-(y_{mn} - \boldsymbol{\beta}^{\dagger}_{1kl} \boldsymbol{x}_{1m} - \boldsymbol{\beta}^{\dagger}_{2kl} \boldsymbol{x}_{2n})^2 \right] - \log \sigma^2_{kl} \right) \Big) \end{gathered} \tag{115}$$

$$\begin{gathered} r_{2nl} \propto \exp\Big(\mathbb{E}_q[\log p_{\psi_2}(\boldsymbol{x}_{2n} | \boldsymbol{\theta}_{2l})] + \Psi(\gamma_{2l}) + \\ \sum_{m=1}^{M} \sum_{k=1}^{K} \frac{r_{1mk}}{\sigma^2_{kl}} \left(\mathbb{E}_q \left[-(y_{mn} - \boldsymbol{\beta}^{\dagger}_{1kl} \boldsymbol{x}_{1m} - \boldsymbol{\beta}^{\dagger}_{2kl} \boldsymbol{x}_{2n})^2 \right] - \log \sigma^2_{kl} \right) \Big) \end{gathered} \tag{116}$$

Maximizing the optimal lower bound with respect to free parameters, the following equations can be used to get their improved estimates:

$$\boldsymbol{\theta}_{1k} = \nabla \psi_1^{-1} \left(\frac{\sum_{m=1}^{M} r_{1mk} (\mathbb{E}_q[\boldsymbol{x}_{1m}])}{\sum_{m=1}^{M} r_{1mk}} \right) \tag{117}$$

$$\boldsymbol{\theta}_{2l} = \nabla \psi_2^{-1} \left(\frac{\sum_{n=1}^{N} r_{2nl} (\mathbb{E}_q[\boldsymbol{x}_{2n}])}{\sum_{n=1}^{N} r_{2nl}} \right) \tag{118}$$

$$\boldsymbol{\beta}_{1kl} = \left[\sum_{m,n=1}^{M,N} r_{1mk} r_{2nl} \mathbb{E}_q[\boldsymbol{x}_{1m} \boldsymbol{x}^{\dagger}_{1m}] \right]^{-1} \left[\sum_{m,n=1}^{M,N} r_{1mk} r_{2nl} (\mathbb{E}_q[y_{mn}] - \boldsymbol{\beta}^{\dagger}_{2kl} \mathbb{E}_q[\boldsymbol{x}_{2n}]) \mathbb{E}_q[\boldsymbol{x}_{1m}] \right] \tag{119}$$

$$\boldsymbol{\beta}_{2kl} = \left[\sum_{m,n=1}^{M,N} r_{1mk} r_{2nl} \mathbb{E}_q[\boldsymbol{x}_{2n} \boldsymbol{x}^{\dagger}_{2n}] \right]^{-1} \left[\sum_{m,n=1}^{M,N} r_{1mk} r_{2nl} (\mathbb{E}_q[y_{mn}] - \boldsymbol{\beta}^{\dagger}_{1kl} \mathbb{E}_q[\boldsymbol{x}_{1m}]) \mathbb{E}_q[\boldsymbol{x}_{2n}] \right] \tag{120}$$

$$\sigma^2_{kl} = \frac{\sum_{m,n=1}^{M,N} r_{1mk} r_{2nl} \left(\mathbb{E}_q \left[(y_{mn} - \boldsymbol{\beta}^{\dagger}_{1kl} \boldsymbol{x}_{1m} - \boldsymbol{\beta}^{\dagger}_{2kl} \boldsymbol{x}_{2n})^2 \right] \right)}{\sum_{m,n=1}^{M,N} r_{1mk} r_{2nl}} \tag{121}$$

$$\boldsymbol{\alpha}_1 = \underset{\boldsymbol{\alpha}_1 \in \mathbb{R}^K_{++}}{\arg\max} \left(\log \frac{\Gamma(\sum_{k=1}^K \alpha_{1k})}{\prod_{k=1}^K \Gamma(\alpha_{1k})} + \sum_{k=1}^K \left(\alpha_{1k} + \sum_{m=1}^M r_{1mk} - 1 \right) \left(\Psi(\gamma_{1k}) - \Psi\left(\sum_{k'=1}^K \gamma_{1k'} \right) \right) \right) \tag{122}$$

$$\boldsymbol{\alpha}_2 = \underset{\boldsymbol{\alpha}_2 \in \mathbb{R}^L_{++}}{\arg\max} \left(\log \frac{\Gamma(\sum_{l=1}^L \alpha_{2l})}{\prod_{l=1}^L \Gamma(\alpha_{2l})} + \sum_{l=1}^L \left(\alpha_{2l} + \sum_{n=1}^N r_{2nl} - 1 \right) \left(\Psi(\gamma_{2l}) - \Psi\left(\sum_{l'=1}^L \gamma_{2l'} \right) \right) \right) \tag{123}$$

Note that the update equations are similar to the updates for LD-AA-BAE framework with the missing entity attributes replaced by the expected values taken with respect to the corresponding variational distributions in equations (109) and (111).

9 Sparse Bayesian Affinity Estimation and Model Selection

This section extends the SABAE framework for learning minimum ℓ_0 norm solutions to generalized linear models for the affinities. The minimum zero-norm solutions result in sparse GLM models distinguishing contributing features from the redundant ones. Traditionally, this has been done by putting a Laplace prior over the coefficients of GLM which is equivalent to penalizing an ℓ_1 norm of the coefficients. The solution to the resulting optimization problem is achieved such that some of the coefficients are zero [31]. However, rather than being dictated by the data, the resulting sparse solution is completely determined by the obtained optimization problem. To overcome this problem, we propose a Sparse Bayesian Affinity Estimation framework (Sp-BAE) that automatically discovers the sparsity structure present in the data. Further, since within the SABAE framework, inherent data heterogeneity is modeled by learning multiple local GLM models, the Sp-BAE framework is able to learn sparsity structure within locally homogeneous partitions of the data. This results in an efficient feature selection framework where the contributions of individual entity attributes towards the modeling of the affinities is automatically learnt in an efficient manner.

The majority of the datasets arising in the domain of affinity relationships are extremely sparse with majority of missing affinities. The resulting data heterogeneity is efficiently modeled within the SABAE framework by simultaneously learning locally homogeneous decompositions of the input space along with the predictive models for the affinities via the use of mixture models. However, the available training data is often too sparse within the local partitions resulting in unreliable predictive models with a limited generalization capability. Hence, in the absence of sufficient training data a trade-off exists for modeling at varying resolutions of the input space. In the later part of this section, we propose an unsupervised model selection framework that automatically learns the resolution of the input space best suited for modeling different partitions of the input space. The resulting framework retains a strong predictive capability for the sparse training data. Note that our definition of model selection is different from the traditional definition prevalent for Bayesian mixture modeling. While in the context of mixture models, model selection refers to an automatic determination of the number of mixture components, we focus on a selection from models at varying resolutions of input affinity space. In the following subsections, we begin with a detailed exposition of the sparse Bayesian affinity estimation framework followed by the model selection for Bayesian affinity estimation.

9.1 Sparse Bayesian Affinity Estimation

In order to achieve sparse solutions within the SABAE framework, we assume that only a subset of the available entity attributes contribute to the prediction of the corresponding affinity relationships. Recall that the affinities are modeled by a mixture generalized linear models in which the affinities are assumed to be drawn from co-cluster specific exponential family distributions whose natural parameter is modeled as a linear combination of the entity attributes, $\boldsymbol{\beta}^{\dagger}_{z_{1m}z_{2n}}\boldsymbol{x}_{mn}$. In Sp-BAE, we assign a Bernoulli random variable b^i_{mn} with each affinity y_{mn} and the term i of the entity attributes such that the random variable assumes a value 1 if the term contributes towards the modeling of the affinity relationship. Hence, the mixture distribution of the affinities can then be written as

$$y_{mn} \sim p_{\psi_{\mathcal{Y}}}(y_{mn}|\boldsymbol{\beta}^{\dagger}_{z_{1m}z_{2n}}(\boldsymbol{b}_{mn} \otimes \boldsymbol{x}_{mn})) \tag{124}$$

where ⊗ denotes element wise product. Such formulation was proposed in the context of sparse linear regression in [35], for language modeling [36] as well as for sparse topic modeling [37]. The sparsity structure can then be efficiently learnt by learning the posterior expected values of these Bernoulli *feature selector* variables. Since, the affinities within a single co-cluster are assumed to be generated from a single GLM distribution, they are expected to follow a common sparsity pattern. Hence, the posterior probability of the selector variables for a specific term is assumed to be shared by affinities within a single co-cluster. Let $\epsilon^i_{z_{1m}z_{2n}}$ be the probability of the Bernoulli random variable b^i_{mn} to assume a value one. Then, the expected sparsity of a co-cluster (k, l) can be defined as follows

$$\mathbb{E}[\text{sparsity}_{kl}|\boldsymbol{\epsilon}_{kl}] \equiv 1 - \sum_{i=1}^{D} \epsilon^i_{kl}/D \tag{125}$$

The *sparsity model* described above can be incorporated into the LD-AA-BAE framework for sparse Bayesian affinity estimation. We next derive a variational EM algorithm for learning the free model parameters including the Bernoulli probabilities $\epsilon^i_{z_{1m}z_{2n}}$. For ease of inference, we assume that the affinities are drawn from a mixture of Gaussian distributions having the following form:

$$y_{mn} \sim N(y_{mn}|\boldsymbol{\beta}^{\dagger}_{z_{1m}z_{2n}}(\boldsymbol{b}_{mn} \otimes \boldsymbol{x}_{mn}), \sigma^2_{z_{1m}z_{2n}})$$

We also assign to each affinity y_{mn}, an additional Bernoulli variable b^0_{mn} which is always one and corresponds to the bias term of the linear model. Hence, $\epsilon^0_{kl} = 1$ for all the co-clusters (k, l). Learning model parameters by an exact EM algorithm requires computation of the observed log-likelihood by marginalization of the latent variables. This requires $KL2^D$ computations for each affinity along with the marginalization of the mixing coefficients. To avoid this expensive computation we introduce a fully factorized mean field approximation to the true posterior distribution of the latent variables.

$$q(\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \mathcal{Y}_{\text{unobs}}, \mathcal{B}, \mathcal{Z}_1, \mathcal{Z}_2) = q^*(\boldsymbol{\pi}_{1m}|\boldsymbol{\gamma}_{1m})q^*(\boldsymbol{\pi}_{2n}|\boldsymbol{\gamma}_{2n})\times \tag{126}$$

$$\left(\prod_{\substack{m,n\\ y_{mn}\in\mathcal{Y}_{\text{unobs}}}} N(y_{mn}|\vartheta_{mn}, \varsigma^2_{mn})\right)\left(\prod_m q^*(z_{1m}|\boldsymbol{r}_{1m})\right)\left(\prod_n q^*(z_{2n}|\boldsymbol{r}_{2n})\right)\left(\prod_{m,n}\prod_i q^*(b^i_{mn}|\delta^i_{mn})\right)$$

The variational distributions for different factors assume a form similar to the ones in LD-AA-BAE model except for the missing affinities which are now approximated by a Gaussian distribution with mean ϑ_{mn} and variance ς^2_{mn}. The variational distribution for the feature selector random variables b^i_{mn} is assumed to be a Bernoulli distribution with variational parameter δ^i_{mn}. Following analysis for the LD-AA-BAE model, a tight lower bound can be constructed over the observed log-likelihood. The variational parameters corresponding to an optimal lower bound then satisfy the following mean field equations:

$$\vartheta_{mn} = \frac{\sum_{k,l=1}^{K,L} \frac{r_{1mk}r_{2nl}}{\sigma^2_{kl}} \left(\sum_{i=0}^{D} \beta^i_{kl}\delta^i_{kl}x^i_{mn}\right)}{\sum_{k,l=1}^{K,L} \frac{r_{1mk}r_{2nl}}{\sigma^2_{kl}}} \tag{127}$$

$$\varsigma_{mn}^2 = \frac{1}{\sum_{k,l=1}^{K,L} \frac{r_{1mk} r_{2nl}}{\sigma_{kl}^2}} \tag{128}$$

$$\gamma_{1k} \quad = \quad \alpha_{1k} + \sum_{m=1}^{M} r_{1mk} \tag{129}$$

$$\gamma_{2l} \quad = \quad \alpha_{2l} + \sum_{n=1}^{N} r_{2nl} \tag{130}$$

$$\delta_{mn}^i = \frac{\exp\left(\sum_{k,l=1}^{K,L} r_{1mk} r_{2nl} \left[\log\left(\frac{\epsilon_{kl}^i}{1-\epsilon_{kl}^i}\right) + \frac{\beta_{kl}^i x_{mn}^i}{2\sigma_{kl}^2}\left(2w_{mn}y_{mn} + 2(1-w_{mn})\vartheta_{mn} - \beta_{kl}^i x_{mn}^i - \sum_{j\neq i} \beta_{kl}^j \delta_{kl}^j x_{mn}^j\right)\right]\right)}{1 + \exp\left(\sum_{k,l=1}^{K,L} r_{1mk} r_{2nl} \left[\log\left(\frac{\epsilon_{kl}^i}{1-\epsilon_{kl}^i}\right) + \frac{\beta_{kl}^i x_{mn}^i}{2\sigma_{kl}^2}\left(2w_{mn}y_{mn} + 2(1-w_{mn})\vartheta_{mn} - \beta_{kl}^i x_{mn}^i - \sum_{j\neq i} \beta_{kl}^j \delta_{kl}^j x_{mn}^j\right)\right]\right)} \tag{131}$$

$$\begin{aligned} r_{1mk} \propto \exp\Bigg(&\log p_{\psi_1}(\boldsymbol{x}_{1m}|\boldsymbol{\theta}_{1k}) + \Psi(\gamma_{1k}) + \sum_{n=1}^{N}\sum_{l=1}^{L} r_{2nl}\Bigg[\sum_{i=0}^{D}\left(\delta_{mn}^i \log \epsilon_{kl}^i + (1-\delta_{mn}^i)\log(1-\epsilon_{kl}^i)\right) - \\ &\frac{1}{2}\log\sigma_{kl}^2 - \frac{1}{2\sigma_{kl}^2}\Bigg(\mathbb{E}_q\Bigg[(y_{mn} - \sum_{i=0}^{D}\beta_{kl}^i b_{mn}^i x_{mn}^i)^2\Bigg]\Bigg)\Bigg]\Bigg) \end{aligned} \tag{132}$$

$$\begin{aligned} r_{2nl} \propto \exp\Bigg(&\log p_{\psi_2}(\boldsymbol{x}_{2n}|\boldsymbol{\theta}_{2l}) + \Psi(\gamma_{2l}) + \sum_{m=1}^{M}\sum_{k=1}^{K} r_{1mk}\Bigg[\sum_{i=0}^{D}\left(\delta_{mn}^i \log \epsilon_{kl}^i + (1-\delta_{mn}^i)\log(1-\epsilon_{kl}^i)\right) - \\ &\frac{1}{2}\log\sigma_{kl}^2 - \frac{1}{2\sigma_{kl}^2}\Bigg(\mathbb{E}_q\Bigg[(y_{mn} - \sum_{i=0}^{D}\beta_{kl}^i b_{mn}^i x_{mn}^i)^2\Bigg]\Bigg)\Bigg]\Bigg) \end{aligned} \tag{133}$$

The coupled mean field equations can be satisfied iteratively to get a tight lower bound on the observed log-likelihood (convergence is guaranteed by convexity of the lower bound). The lower bound can then be maximized with respected to the free model parameters to obtain an improved estimate of the parameters.

$$\boldsymbol{\theta}_{1k} \quad = \quad \nabla\psi_1^{-1}\left(\frac{\sum_{m=1}^{M} r_{1mk}\boldsymbol{x}_{1m}}{\sum_{m=1}^{M} r_{1mk}}\right) \tag{134}$$

$$\boldsymbol{\theta}_{2l} \quad = \quad \nabla\psi_2^{-1}\left(\frac{\sum_{n=1}^{N} r_{2nl}\boldsymbol{x}_{2n}}{\sum_{n=1}^{N} r_{2nl}}\right) \tag{135}$$

$$\boldsymbol{\alpha}_1 \quad = \quad \underset{\boldsymbol{\alpha}_1\in\mathbb{R}_{++}^K}{\arg\max}\left(\log\frac{\Gamma(\sum_{k=1}^{K}\alpha_{1k})}{\prod_{k=1}^{K}\Gamma(\alpha_{1k})} + \sum_{k=1}^{K}\left(\alpha_{1k} + \sum_{m=1}^{M} r_{1mk} - 1\right)\left(\Psi(\gamma_{1k}) - \Psi\left(\sum_{k'=1}^{K}\gamma_{1k'}\right)\right)\right) \tag{136}$$

$$\boldsymbol{\alpha}_2 \quad = \quad \underset{\boldsymbol{\alpha}_2\in\mathbb{R}_{++}^L}{\arg\max}\left(\log\frac{\Gamma(\sum_{l=1}^{L}\alpha_{2l})}{\prod_{l=1}^{L}\Gamma(\alpha_{2l})} + \sum_{l=1}^{L}\left(\alpha_{2l} + \sum_{n=1}^{N} r_{2nl} - 1\right)\left(\Psi(\gamma_{2l}) - \Psi\left(\sum_{l'=1}^{L}\gamma_{2l'}\right)\right)\right) \tag{137}$$

$$\epsilon_{kl}^i = \frac{\sum_{m,n=1}^{M,N} r_{1mk} r_{2nl}\delta_{mn}^i}{\sum_{m,n=1}^{M,N} r_{1mk} r_{2nl}} \tag{138}$$

$$\beta^i_{kl} = \frac{\sum_{m,n=1}^{M,N} r_{1mk} r_{2nl} \delta^i_{mn} \left(w_{mn} y_{mn} + (1 - w_{mn}) \vartheta_{mn} - \sum_{j \neq i} \beta^j_{kl} \delta^j_{mn} x^j_{mn} \right)}{\sum_{m,n=1}^{M,N} r_{1mk} r_{2nl} \delta^i_{mn}} \tag{139}$$

The update equations are similar to M-step update equations of LD-AA-BAE. Since a Gaussian distribution assumption is assumed over the affinities, a closed form expression is obtained for the updates of the GLM coefficients $\boldsymbol{\beta}_{kl}$. Further, each term i of the entity attributes is weighted by posterior feature selector probability yielding the desired sparsity structure.

9.2 From Local to Global: Model Selection for Bayesian Affinity Estimation

Using the basic SABAE framework, the modeling of affinity relationship between a pair of entities can be achieved at following resolutions of the input affinity space

1. *Local Modeling* corresponds to modeling the affinities by a mixture model with separate GLM parameters for each co-cluster. This requires learning of KL models, one for each co-cluster.

$$y_{mn} \sim p_{\psi_y}(y_{mn} | \boldsymbol{\beta}^{\dagger}_{1 z_{1m} z_{2n}} \boldsymbol{x}_{mn}) \tag{140}$$

2. *Shrinked Modeling* retains separate models for the clusters (in place of co-clusters) for the two identity sets resulting in $K + L$ models, one for each cluster. The affinities within a co-cluster are modeled by borrowing the parameters of the corresponding clusters.

$$y_{mn} \sim p_{\psi_y}(y_{mn} | \boldsymbol{\beta}_{2 z_{1m}} \boldsymbol{x}_{1m} + \boldsymbol{\beta}^{\dagger}_{2 z_{2n}} \boldsymbol{x}_{2n}) \tag{141}$$

3. *Global Modeling* corresponds to modeling the affinities by a single global model and corresponds to a model with fewest parameters.

$$y_{mn} \sim p_{\psi_y}(y_{mn} | \boldsymbol{\beta}^{\dagger}_{3} \boldsymbol{x}_{mn}) \tag{142}$$

In the presence of sufficient amount of training data, local modeling approach can efficiently capture complex affinity relationship structures in the input space by discovering homogeneous partitions of the data leveraging the flexibility provided by separate model for each co-cluster. However, if the training data is sparse, this flexibility results in the models being overfitted to the given data resulting in over-training and hence a limited generalization capability. On the extreme, one can utilize a single global model to overcome this problem for sparse training data. However, a single global model fails to capture the complex interactions between entities that are important for describing the resulting affinity relationships. To avoid this extreme behavior associated with local and global models, one can follow a shrinked methodology by sharing parameters across individual clusters as proposed in [13].

Often, the affinity relationships and hence the resulting heterogeneity structures are very complex such that a single modeling assumption is unable to account for the complexity resulting in a sub-optimal performance. What is needed is thus, an automated model selection framework that chooses the modeling assumption that best describes the underlying heterogeneity structure. We propose such a model selection framework that assumes the affinities to be generated from a *mixture* of the three possible modeling choices

$$y_{mn} \sim \epsilon_{1mn} p_{\psi_y}(y_{mn} | \boldsymbol{\beta}^{\dagger}_{1 z_{1m} z_{2n}} \boldsymbol{x}_{mn}) + \epsilon_{2mn} p_{\psi_y}(y_{mn} | \boldsymbol{\beta}_{2 z_{1m}} \boldsymbol{x}_{1m} + \boldsymbol{\beta}^{\dagger}_{2 z_{2n}} \boldsymbol{x}_{2n}) + \epsilon_{3mn} p_{\psi_y}(y_{mn} | \boldsymbol{\beta}^{\dagger}_{3} \boldsymbol{x}_{mn}) \tag{143}$$

where the mixing coefficients ϵ_{mn} are the probabilities of each modeling choice. Hence, within a Bayesian framework one can easily learn these mixing coefficients to automatically determine the best modeling assumption for each affinity. For a co-cluster with sufficient number of training affinities where a reliable local model can be learnt, the above model selection framework reflects this property by assigning a high value to the corresponding mixing coefficient ϵ_{1mn}. Similarly, varying proportions of the coefficients enable modeling of varying levels of heterogeneity in the co-clusters. We extend the LD-AA-BAE framework for such a model selection task by assuming the affinities to be generated from the mixture distribution described in (143). Further, we constrain the affinities within the same co-cluster to have similar mixture distribution by sharing the mixing coefficients ϵ_{mn} in the co-cluster.

The model parameters can be learnt using a variational EM algorithm. The following variational distribution can be assumed over the latent variables

$$q(\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \mathcal{Y}_{\text{unobs}}, \mathcal{Z}_1, \mathcal{Z}_2) = \tag{144}$$

$$q^*(\boldsymbol{\pi}_{1m}|\boldsymbol{\gamma}_{1m}) q^*(\boldsymbol{\pi}_{2n}|\boldsymbol{\gamma}_{2n}) \left(\prod_{\substack{m,n \\ y_{mn} \in \mathcal{Y}_{\text{unobs}}}} q^*_{\psi_y}(y_{mn}|\phi_{mn}) \right) \left(\prod_m q^*(z_{1m}|\boldsymbol{r}_{1m}) \right) \left(\prod_n q^*(z_{2n}|\boldsymbol{r}_{2n}) \right)$$

Similar to the mean field updates for the LD-AA-BAE model, the following mean field equations can be derieved for the variational parameters (For the basic methodology, refer to variational EM algorithm derivation for LD-AA-BAE model, section 3.2):

$$\phi_{mn} = \sum_{k,l=1}^{K,L} r_{1mk} r_{2nl} \left(\epsilon_{1kl} \boldsymbol{\beta}_{1kl}^{\dagger} \boldsymbol{x}_{mn} + \epsilon_{2kl} \left(\boldsymbol{\beta}_{1k}^{\dagger} \boldsymbol{x}_{1m} + \boldsymbol{\beta}_{2l}^{\dagger} \boldsymbol{x}_{2n} \right) + \epsilon_{3kl} \boldsymbol{\beta}_{3}^{\dagger} \boldsymbol{x}_{mn} \right) \tag{145}$$

$$\gamma_{1k} = \alpha_{1k} + \sum_{m=1}^{M} r_{1mk} \tag{146}$$

$$\gamma_{2l} = \alpha_{2l} + \sum_{n=1}^{N} r_{2nl} \tag{147}$$

$$r_{1mk} \propto \exp\Big(\log p_{\psi_1}(\boldsymbol{x}_{1m}|\boldsymbol{\theta}_{1k}) + \Psi(\gamma_{1k}) +$$

$$\sum_{n=1}^{N} \sum_{l=1}^{L} r_{2nl} \left(\mathbb{E}_q \left[\epsilon_{1kl} \log p_{\psi_y}(y_{mn}|\boldsymbol{\beta}_{1kl}^{\dagger} \boldsymbol{x}_{mn}) + \epsilon_{2kl} \log p_{\psi_y}(y_{mn}|\boldsymbol{\beta}_{1k}^{\dagger} \boldsymbol{x}_{1m} + \boldsymbol{\beta}_{2l}^{\dagger} \boldsymbol{x}_{2n}) + \epsilon_{3kl} \log p_{\psi_y}(y_{mn}|\boldsymbol{\beta}_{3}^{\dagger} \boldsymbol{x}_{mn}) \right] \right) \Bigg) \tag{148}$$

$$r_{2nl} \propto \exp\Big(\log p_{\psi_2}(\boldsymbol{x}_{2n}|\boldsymbol{\theta}_{2l}) + \Psi(\gamma_{2l}) +$$

$$\sum_{m=1}^{M} \sum_{k=1}^{K} r_{1mk} \left(\mathbb{E}_q \left[\epsilon_{1kl} \log p_{\psi_y}(y_{mn}|\boldsymbol{\beta}_{1kl}^{\dagger} \boldsymbol{x}_{mn}) + \epsilon_{2kl} \log p_{\psi_y}(y_{mn}|\boldsymbol{\beta}_{1k}^{\dagger} \boldsymbol{x}_{1m} + \boldsymbol{\beta}_{2l}^{\dagger} \boldsymbol{x}_{2n}) + \epsilon_{3kl} \log p_{\psi_y}(y_{mn}|\boldsymbol{\beta}_{3}^{\dagger} \boldsymbol{x}_{mn}) \right] \right) \Bigg) \tag{149}$$

The update equations for the free model parameters can similarly be obtained by maximizing the resulting optimal lower bound from the mean field updates.

$$\boldsymbol{\theta}_{1k} = \nabla \psi_1^{-1} \left(\frac{\sum_{m=1}^{M} r_{1mk} \boldsymbol{x}_{1m}}{\sum_{m=1}^{M} r_{1mk}} \right) \tag{150}$$

$$\boldsymbol{\theta}_{2l} = \nabla \psi_2^{-1} \left(\frac{\sum_{n=1}^{N} r_{2nl} \boldsymbol{x}_{2n}}{\sum_{n=1}^{N} r_{2nl}} \right) \tag{151}$$

$$\boldsymbol{\alpha}_1 = \underset{\boldsymbol{\alpha}_1 \in \mathbb{R}^K_{++}}{\arg\max} \left(\log \frac{\Gamma(\sum_{k=1}^{K} \alpha_{1k})}{\prod_{k=1}^{K} \Gamma(\alpha_{1k})} + \sum_{k=1}^{K} \left(\alpha_{1k} + \sum_{m=1}^{M} r_{1mk} - 1 \right) \left(\Psi(\gamma_{1k}) - \Psi\left(\sum_{k'=1}^{K} \gamma_{1k'} \right) \right) \right) \tag{152}$$

$$\boldsymbol{\alpha}_2 = \underset{\boldsymbol{\alpha}_2 \in \mathbb{R}^L_{++}}{\arg\max} \left(\log \frac{\Gamma(\sum_{l=1}^{L} \alpha_{2l})}{\prod_{l=1}^{L} \Gamma(\alpha_{2l})} + \sum_{l=1}^{L} \left(\alpha_{2l} + \sum_{n=1}^{N} r_{2nl} - 1 \right) \left(\Psi(\gamma_{2l}) - \Psi\left(\sum_{l'=1}^{L} \gamma_{2l'} \right) \right) \right) \tag{153}$$

The parameters of the three modeling choices can be updated in the M-step using solutions to the following optimization problem:

$$\boldsymbol{\beta}_{1kl} = \underset{\boldsymbol{\beta}_{1kl} \in \mathbb{R}^D}{\arg\max} \sum_{m,n=1}^{M,N} r_{1mk} r_{2nl} \epsilon_{1kl} \left(\left\langle (w_{mn} y_{mn} + (1 - w_{mn}) \nabla \psi_{y}(\phi_{mn})), \boldsymbol{\beta}^{\dagger}_{1kl} \boldsymbol{x}_{mn} \right\rangle - \psi_{y} \left(\boldsymbol{\beta}^{\dagger}_{1kl} \boldsymbol{x}_{mn} \right) \right) \tag{154}$$

$$\boldsymbol{\beta}_{2k} = \underset{\boldsymbol{\beta}_{2k} \in \mathbb{R}^{D_1}}{\arg\max} \sum_{m,n=1}^{M,N} \sum_{l=1}^{L} r_{1mk} r_{2nl} \epsilon_{2kl} \left(\left\langle (w_{mn} y_{mn} + (1 - w_{mn}) \nabla \psi_{y}(\phi_{mn})), \boldsymbol{\beta}^{\dagger}_{2k} \boldsymbol{x}_{1m} \right\rangle - \psi_{y} \left(\boldsymbol{\beta}^{\dagger}_{1k} \boldsymbol{x}_{1m} + \boldsymbol{\beta}^{\dagger}_{2l} \boldsymbol{x}_{2n} \right) \right) \tag{155}$$

$$\boldsymbol{\beta}_{2l} = \underset{\boldsymbol{\beta}_{2l} \in \mathbb{R}^{D_2}}{\arg\max} \sum_{m,n=1}^{M,N} \sum_{k=1}^{K} r_{1mk} r_{2nl} \epsilon_{2kl} \left(\left\langle (w_{mn} y_{mn} + (1 - w_{mn}) \nabla \psi_{y}(\phi_{mn})), \boldsymbol{\beta}^{\dagger}_{2l} \boldsymbol{x}_{2n} \right\rangle - \psi_{y} \left(\boldsymbol{\beta}^{\dagger}_{1k} \boldsymbol{x}_{1m} + \boldsymbol{\beta}^{\dagger}_{2l} \boldsymbol{x}_{2n} \right) \right) \tag{156}$$

$$\boldsymbol{\beta}_{3} = \underset{\boldsymbol{\beta}_{3} \in \mathbb{R}^{D}}{\arg\max} \sum_{m,n=1}^{M,N} \sum_{k,l=1}^{K,L} r_{1mk} r_{2nl} \epsilon_{3kl} \left(\left\langle (w_{mn} y_{mn} + (1 - w_{mn}) \nabla \psi_{y}(\phi_{mn})), \boldsymbol{\beta}^{\dagger}_{3} \boldsymbol{x}_{mn} \right\rangle - \psi_{y} \left(\boldsymbol{\beta}^{\dagger}_{3} \boldsymbol{x}_{mn} \right) \right) \tag{157}$$

Closed form expressions can be derived for updating the mixture coefficients corresponding to each modeling choice that assume the following forms:

$$\epsilon_{1kl} = \frac{\sum_{m,n=1}^{M,N} r_{1mk} r_{2nl} \mathbb{E}_q \left[\log p_{\psi_{y}}(y_{mn} | \boldsymbol{\beta}^{\dagger}_{1kl} \boldsymbol{x}_{mn}) \right]}{\sum_{m,n=1}^{M,N} r_{1mk} r_{2nl} \left(\mathbb{E}_q \left[\log p_{\psi_{y}}(y_{mn} | \boldsymbol{\beta}^{\dagger}_{1kl} \boldsymbol{x}_{mn}) + \log p_{\psi_{y}}(y_{mn} | \boldsymbol{\beta}^{\dagger}_{1k} \boldsymbol{x}_{1m} + \boldsymbol{\beta}^{\dagger}_{2l} \boldsymbol{x}_{2n}) + \log p_{\psi_{y}}(y_{mn} | \boldsymbol{\beta}^{\dagger}_{3} \boldsymbol{x}_{mn}) \right] \right)} \tag{158}$$

$$\epsilon_{2kl} = \frac{\sum_{m,n=1}^{M,N} r_{1mk} r_{2nl} \mathbb{E}_q \left[\log p_{\psi_{y}}(y_{mn} | \boldsymbol{\beta}^{\dagger}_{1k} \boldsymbol{x}_{1m} + \boldsymbol{\beta}^{\dagger}_{2l} \boldsymbol{x}_{2n}) \right]}{\sum_{m,n=1}^{M,N} r_{1mk} r_{2nl} \left(\mathbb{E}_q \left[\log p_{\psi_{y}}(y_{mn} | \boldsymbol{\beta}^{\dagger}_{1kl} \boldsymbol{x}_{mn}) + \log p_{\psi_{y}}(y_{mn} | \boldsymbol{\beta}^{\dagger}_{1k} \boldsymbol{x}_{1m} + \boldsymbol{\beta}^{\dagger}_{2l} \boldsymbol{x}_{2n}) + \log p_{\psi_{y}}(y_{mn} | \boldsymbol{\beta}^{\dagger}_{3} \boldsymbol{x}_{mn}) \right] \right)} \tag{159}$$

$$\epsilon_{3kl} = \frac{\sum_{m,n=1}^{M,N} r_{1mk} r_{2nl} \mathbb{E}_q \left[\log p_{\psi_{y}}(y_{mn} | \boldsymbol{\beta}^{\dagger}_{3} \boldsymbol{x}_{mn}) \right]}{\sum_{m,n=1}^{M,N} r_{1mk} r_{2nl} \left(\mathbb{E}_q \left[\log p_{\psi_{y}}(y_{mn} | \boldsymbol{\beta}^{\dagger}_{1kl} \boldsymbol{x}_{mn}) + \log p_{\psi_{y}}(y_{mn} | \boldsymbol{\beta}^{\dagger}_{1k} \boldsymbol{x}_{1m} + \boldsymbol{\beta}^{\dagger}_{2l} \boldsymbol{x}_{2n}) + \log p_{\psi_{y}}(y_{mn} | \boldsymbol{\beta}^{\dagger}_{3} \boldsymbol{x}_{mn}) \right] \right)} \tag{160}$$

Thus, by explicitly learning the mixing coefficients, one can efficiently learn the contribution of each choice for a specific co-cluster.

10 Learning to Rank Affinities

In many applications, the learnt affinities are used to generate a preference list over one set of entities for an entity of the other. For example, the central goal in most collaborative filtering applications is to make *top-k* recommendations to different users. This is often done by first estimating user specific affinities for a given set of items, following which a preference list is generated by ranking the estimated affinities. As such, the bulk of the effort is spent in accurately estimating the missing affinities [38]. However, to generate a preference list one only needs to learn an ordering on the missing affinities rather than the actual values of the affinities themselves. This section introduces a supervised ranking model that efficiently learns such an ordering. In particular, the model learns pair wise ordering of affinities as a function of entity ('user' and 'item') attributes allowing efficient generation of preference lists.

We consider a supervised ranking setting in which each training example consists of a *query*, a set of input *results* and a (partial) *preference* over the results. A query result pair (q, r) is characterized by attributes $\boldsymbol{x}_{qr}$. The learning task is to discover a function, known as a *scoring function* that provides a query specific ordering of inputs that best respects the observed preferences. Generally, the scoring function is a parametric function of the attributes, $f(\boldsymbol{x}_{qr}; \boldsymbol{\theta})$ and learning entails estimating the parameters $\boldsymbol{\theta}$. For example, in a movie recommendation engine, each query is a specific user for whom we wish to generate a preference list over a set of movies (results). A partial ordering of the movies is obtained from the observed ratings while the attributes can be obtained from covariates associated with the user and the movies. For rest of the section, we describe our ranking models using the movie recommendation example. However, it should be noted that the models are generic in their applicability to similar affinity ranking problems.

10.1 Supervised Ranking

We start with a brief introduction to supervised ranking using a movie recommendation example for exposition of the basic ideas. In subsequent sections, we show how the basic supervised setting can be enhanced to capture the dyadic property common to many affinity recording datasets, allowing generation of more entity-specific preference lists.

Let M be the number of users and N be the number of movies on which we intend to generate a ranking for each individual user. A set of observed ratings can be represented as particular entries of an $M \times N$ matrix $\boldsymbol{Y} = \{y_{mi}\}, [m]_1^M, [i]_1^N$. To distinguish observed entries from unobserved entries in the matrix, we assign a weight $w_{mi} \in \{0, 1\}$ to each rating y_{im} such that $w_{mi} = 1$ if y_{mi} is observed and 0 otherwise. Following [38], the supervised ranking problem can then be formulated as a minimization of a conditional surrogate loss of the following form:

$$\varphi(\boldsymbol{\theta}) = \mathbb{E}_R \left[\sum_{i,j \in R} h(a_{ij}^R) \phi(f(\boldsymbol{x}_j; \boldsymbol{\theta}) - f(\boldsymbol{x}_i; \boldsymbol{\theta})) \right] \tag{161}$$

The loss is a weighted disagreement cost incurred when the scoring functions, f for the entities i, j in the given preference ranking list R disagree with the given order. The incurred disagreement cost is (a_{ij}^R), where h is a function of the penalties a_{ij}^R and $\phi \geq 0$ is a non-increasing function. It was shown in [38], for ϕ convex, the loss defined in (161) fails to asymptotically minimize the Bayes risk and hence is inconsistent. However, it was shown that under certain conditions a regularized linear loss of the following form is asymptotically consistent:

$$\varphi(\boldsymbol{\theta}) = \mathbb{E}_R \left[\sum_{i,j \in R} h(a_{ij}^R)(f(\boldsymbol{x}_j; \boldsymbol{\theta}) - f(\boldsymbol{x}_i; \boldsymbol{\theta})) + \upsilon \sum_i r(f(\boldsymbol{x}_i; \boldsymbol{\theta})) \right] \tag{162}$$

where $\upsilon > 0$ and r is strictly convex and 1-coercive. Further, the required conditions for consistency of the linear loss in (162) are satisfied if the penalties $h(a_{ij})$ assume the following form [39]:

$$h(a_{ij}) = s_i - s_j, \tag{163}$$

where s_i is a score associated with the entity $\boldsymbol{x}_i$ to be ranked. Hence, for the movie recommendation engine, the supervised ranking loss can be written as

$$\varphi(\boldsymbol{\theta}) = \sum_{m=1}^{M}\sum_{i,j=1}^{N} w_{mi}w_{mj}(y_{mi} - y_{mj})(f(\boldsymbol{x}_{mj};\boldsymbol{\theta}) - f(\boldsymbol{x}_{mi};\boldsymbol{\theta})) + \upsilon \sum_{m=1}^{M}\sum_{i=1}^{N} w_{mi} r(f(\boldsymbol{x}_{mi};\boldsymbol{\theta})) \tag{164}$$

For remainder of the section, we concentrate on the loss given by (164) and develop efficient forms of the scoring function $f(\boldsymbol{x};\boldsymbol{\theta})$ that capture the dyadic nature of the data arising in these domains.

10.2 Ranking Affinities

A valid form for the scoring function f consists of a parameterized model over the query-input features. For our case, a query corresponds to a user while the inputs are the sets of movies for which the ranking needs to be generated. To efficiently capture the attributes associated with a user-movie pair, we assume that the scoring function is a bi-linear model over the user-movie attributes [31]. Also, to include the user (movie) specific biases which are an important property of datasets arising in such domains, we include a factorization term comprising of a cross-product between user and movie specific factors. The resulting parameterized scoring function is then obtained as follows:

$$f(\boldsymbol{x}_{mi};\boldsymbol{\Gamma},\boldsymbol{u},\boldsymbol{v}) = \boldsymbol{x}_{1m}^{\dagger}\boldsymbol{\Gamma}\boldsymbol{x}_{2i} + \boldsymbol{u}_m^{\dagger}\boldsymbol{v}_i \tag{165}$$

Using (164), the model parameters $(\boldsymbol{\Gamma},\boldsymbol{u},\boldsymbol{v})$ can then be learnt by minimizing the following loss function

$$\begin{aligned}\varphi(\boldsymbol{\Gamma},\boldsymbol{u},\boldsymbol{v}) = &\sum_{m=1}^{M}\sum_{i,j=1}^{N} w_{mi}w_{mj}(y_{mi} - y_{mj})\left[\boldsymbol{x}_{1m}^{\dagger}\boldsymbol{\Gamma}(\boldsymbol{x}_{2j} - \boldsymbol{x}_{2i}) + \boldsymbol{u}_m^{\dagger}(\boldsymbol{v}_j - \boldsymbol{v}_i)\right] + \upsilon \sum_{m=1}^{M}\sum_{i=1}^{N} w_{mi} r(\boldsymbol{x}_{1m}^{\dagger}\boldsymbol{\Gamma}\boldsymbol{x}_{2i} + \boldsymbol{u}_m^{\dagger}\boldsymbol{v}_i)\\ &+\frac{\lambda}{2}\left(\|\boldsymbol{\Gamma}\|_F^2 + \sum_{m=1}^{M}\|\boldsymbol{u}_m\|_2^2 + \sum_{i=1}^{N}\|\boldsymbol{v}_i\|_2^2\right)\end{aligned} \tag{166}$$

wherein we have included *ℓ-2* regularizers on the model parameters. Since r is strictly convex, the loss function can be efficiently minimized using stochastic gradient descent. The following equations provide the resulting gradient expressions for each of the parameters:

$$\begin{aligned}\frac{\partial\varphi}{\partial\boldsymbol{\Gamma}} &= \sum_{m=1}^{M}\sum_{i,j=1}^{N} w_{mi}w_{mj}(y_{mi} - y_{mj})\left[\boldsymbol{x}_{1m}(\boldsymbol{x}_{2j} - \boldsymbol{x}_{2i})^{\dagger}\right] + \upsilon \sum_{m=1}^{M}\sum_{i=1}^{N} w_{mi}\left[\nabla_{\boldsymbol{\Gamma}} r(\boldsymbol{x}_{1m}^{\dagger}\boldsymbol{\Gamma}\boldsymbol{x}_{2i} + \boldsymbol{u}_m^{\dagger}\boldsymbol{v}_i)\right] + \lambda\boldsymbol{\Gamma}\\ \frac{\partial\varphi}{\partial\boldsymbol{u}_m} &= \sum_{i,j=1}^{N} w_{mi}w_{mj}(y_{mi} - y_{mj})\left[(\boldsymbol{v}_j - \boldsymbol{v}_i)\right] + \upsilon \sum_{i=1}^{N} w_{mi}\left[\nabla_{\boldsymbol{u}_m} r(\boldsymbol{x}_{1m}^{\dagger}\boldsymbol{\Gamma}\boldsymbol{x}_{2i} + \boldsymbol{u}_m^{\dagger}\boldsymbol{v}_i)\right] + \lambda\boldsymbol{u}_m\\ \frac{\partial\varphi}{\partial\boldsymbol{v}_i} &= \sum_{m=1}^{M}\sum_{j=1}^{N} w_{mi}w_{mj}(y_{mi} - y_{mj})\left[-\boldsymbol{u}_m\right] + \upsilon \sum_{m=1}^{M} w_{mi}\left[\nabla_{\boldsymbol{v}_i} r(\boldsymbol{x}_{1m}^{\dagger}\boldsymbol{\Gamma}\boldsymbol{x}_{2i} + \boldsymbol{u}_m^{\dagger}\boldsymbol{v}_i)\right] + \lambda\boldsymbol{v}_i\end{aligned}$$

Note that the scoring function consists of a *global term* $\boldsymbol{x}_{1m}^{\dagger}\boldsymbol{\Gamma}\boldsymbol{x}_{2i}$, wherein a common $\boldsymbol{\Gamma}$ is shared for every user movie pair, (m, i). Similarly, the function also contains a *local term* $\boldsymbol{u}_m^{\dagger}\boldsymbol{v}_i$ such that the parameters $\boldsymbol{u}_m$ and $\boldsymbol{v}_i$ are different for each user m and movie i respectively. However, such a parameterization on the two extremes fails to leverage the inherent heterogeneity brought in by the dyadic nature of such datasets. What is needed is a *smooth* parameter sharing mechanism in between the two extremes. SABAE framework provides an efficient backdrop for such a smooth parameter sharing. Since we are utilizing a conditional surrogate loss minimization within a decision theoretic framework for ranking affinities, a *hard assignment* version of SABAE can be used to partition the data matrix into a grid of blocks or co-clusters. The different parameters can then be shared across users and movies within each co-cluster. Each co-cluster (k, l) is associated with model parameters $(\mathbf{\Gamma}_{kl}, \boldsymbol{u}_k, \boldsymbol{v}_l)$ that are shared by the users and movies that belong to the co-cluster. However, a separate set of model parameters for every co-cluster can lead to severe overfitting when the data is very sparse that is usually the case in such domains. To overcome this issue, one can assume a rank t approximation on the bi-linear model of the form $\mathbf{\Gamma}_{kl} = \mathbf{\Gamma}_{1k}\mathbf{\Gamma}_{2l}^{\dagger}$.

Let ρ be a mapping from the M users to the K user clusters and γ be a mapping from the N movies to the L movie clusters. The co-clustering assignments (ρ, γ) and the shared co-cluster parameters $(\mathbf{\Gamma}_{1k}, \mathbf{\Gamma}_{2l}, \boldsymbol{u}_k, \boldsymbol{v}_l)$, $[k]_1^K$, $[l]_1^L$ can be efficiently learnt by minimizing the following loss function

$$\varphi(\mathbf{\Gamma}_1, \mathbf{\Gamma}_2, \boldsymbol{u}, \boldsymbol{v}) = \sum_{k=1}^{K} \sum_{l_1,l_2 \in C} \sum_{\substack{m:\rho(m)=k \\ i,j:\gamma(i),\gamma(j)\in\{l_1,l_2\}}} w_{mi} w_{mj} (y_{mi} - y_{mj}) \left[\boldsymbol{x}_{1m}^{\dagger} \mathbf{\Gamma}_{1k} \left(\mathbf{\Gamma}_{2\gamma(j)}^{\dagger} \boldsymbol{x}_{2j} - \boldsymbol{\beta}_{2\gamma(i)}^{\dagger} \boldsymbol{x}_{2i}\right) + \boldsymbol{u}_k^{\dagger} \left(\boldsymbol{v}_{\gamma(j)} - \boldsymbol{v}_{\gamma(i)}\right)\right] +$$

$$\upsilon \sum_{k=1}^{K} \sum_{l=1}^{L} \sum_{\substack{m:\rho(m)=k \\ i:\gamma(i)=l}} w_{mi} r(\boldsymbol{x}_{1m}^{\dagger} \mathbf{\Gamma}_{1k} \mathbf{\Gamma}_{2l}^{\dagger} \boldsymbol{x}_{2i} + \boldsymbol{u}_k^{\dagger} \boldsymbol{v}_l) + \frac{\lambda}{2} \left(\sum_{k=1}^{K} \parallel \mathbf{\Gamma}_{1k} \parallel_F^2 + \sum_{l=1}^{L} \parallel \mathbf{\Gamma}_{2l} \parallel_F^2 + \sum_{k=1}^{K} \parallel \boldsymbol{u}_k \parallel_2^2 + \sum_{l=1}^{L} \parallel \boldsymbol{v}_l \parallel_2^2 \right) \tag{167}$$

The set C is a set of all possible $\binom{L}{2} + L$ movie cluster index pairs of the form (l_1, l_2). The loss function can be efficiently minimized using an iterative procedure to simultaneously learn the model parameters and the co-clustering assignments [13]. Beginning with a random clustering assignments, the model parameters are first learnt by gradient descent following which each user (movie) is assigned to a user (movie) cluster that minimizes the loss function in (167). The two steps that directly minimize the loss function guarantee convergence to local minima. The expression for the gradient with respect to each paramter is as follows:

$$\frac{\partial \varphi}{\partial \mathbf{\Gamma}_{1k}} = \sum_{l_1,l_2 \in C} \sum_{\substack{m:\rho(m)=k \\ i,j:\gamma(i),\gamma(j)\in\{l_1,l_2\}}} w_{mi} w_{mj} (y_{mi} - y_{mj}) \left[\boldsymbol{x}_{1m}^{\dagger} \left(\mathbf{\Gamma}_{2\gamma(j)}^{\dagger} \boldsymbol{x}_{2j} - \mathbf{\Gamma}_{2\gamma(i)}^{\dagger} \boldsymbol{x}_{2i}\right)\right] +$$

$$\upsilon \sum_{l=1}^{L} \sum_{\substack{m:\rho(m)=k \\ i:\gamma(i)=l}} w_{mi} \left[\nabla_{\mathbf{\Gamma}_{1k}} r(\boldsymbol{x}_{1m}^{\dagger} \mathbf{\Gamma}_{1k} \mathbf{\Gamma}_{2l}^{\dagger} \boldsymbol{x}_{2i} + \boldsymbol{u}_k^{\dagger} \boldsymbol{v}_l)\right] + \lambda \mathbf{\Gamma}_{1k}$$

$$\frac{\partial \varphi}{\partial \mathbf{\Gamma}_{2l}} = \sum_{k=1}^{K} \sum_{\substack{m:\rho(m)=k \\ i:\gamma(i)=l}} w_{mi} w_{mj} (y_{mi} - y_{mj}) \left[-\boldsymbol{x}_{2i} \boldsymbol{x}_{1m}^{\dagger} \mathbf{\Gamma}_{1k}\right] + \upsilon \sum_{k=1}^{K} \sum_{\substack{m:\rho(m)=k \\ i:\gamma(i)=l}} w_{mi} \left[\nabla_{\mathbf{\Gamma}_{2l}} r(\boldsymbol{x}_{1m}^{\dagger} \mathbf{\Gamma}_{2l} \mathbf{\Gamma}_{2l}^{\dagger} \boldsymbol{x}_{2i} + \boldsymbol{u}_k^{\dagger} \boldsymbol{v}_l)\right] + \lambda \mathbf{\Gamma}_{2l}$$

$$\frac{\partial \varphi}{\partial \boldsymbol{u}_k} = \sum_{l_1,l_2 \in C} \sum_{\substack{m:\rho(m)=k \\ i,j:\gamma(i),\gamma(j)\in\{l_1,l_2\}}} w_{mi} w_{mj} (y_{mi} - y_{mj}) \left[(\boldsymbol{u}_{\gamma(j)} - \boldsymbol{u}_{\gamma(i)})\right] + \upsilon \sum_{l=1}^{L} \sum_{\substack{m:\rho(m)=k \\ i:\gamma(i)=l}} w_{mi} \left[\nabla_{\boldsymbol{u}_k} r(\boldsymbol{x}_{1m}^{\dagger} \mathbf{\Gamma}_{2l} \mathbf{\Gamma}_{2l}^{\dagger} \boldsymbol{x}_{2i} + \boldsymbol{u}_k^{\dagger} \boldsymbol{v}_l)\right] + \lambda \boldsymbol{u}_k$$

$$\frac{\partial \varphi}{\partial \boldsymbol{v}_l} = \sum_{k=1}^{K} \sum_{\substack{m:\rho(m)=k \\ i:\gamma(i)=l}} w_{mi} w_{mj} (y_{mi} - y_{mj}) \left[-\boldsymbol{v}_k \right] + \upsilon \sum_{k=1}^{K} \sum_{\substack{m:\rho(m)=k \\ i:\gamma(i)=l}} w_{mi} \left[\nabla_{\boldsymbol{v}_l} r(\boldsymbol{x}_{1m}^{\dagger} \boldsymbol{\beta}_{2l} \boldsymbol{\beta}_{2l}^{\dagger} \boldsymbol{x}_{2i} + \boldsymbol{u}_k^{\dagger} \boldsymbol{v}_l) \right] + \lambda \boldsymbol{v}_l$$

11 Semi-supervised Co-clustering and Matrix Approximation

This section extends the NA-BAE framework to address the problem of co-clustering a data matrix in the presence of side information on clustered entities. In particular, we solve the problem of co-clustering a matrix with potentially large number of missing entries into a grid of blocks when some *neighborhood* information is available for individual rows and columns. Similar to NA-BAE, the neighborhood information is encoded using a markov random field prior on latent membership variables. The use of the neighborhood information helps to address new rows and columns which the traditional co-clustering methods fail to account for. A Bayesian approach helps to estimate the missing entries of the matrix as a side product.

We intend to co-cluster an $M \times N$ data matrix into $K \times L$ blocks formed by an intersection of K row clusters and L column clusters. The matrix has a potentially large number of missing entries represented by the set $\mathcal{Y}_{\text{unobs}}$ and a set of few known entries $\mathcal{Y}_{\text{obs}}$. The set of all $M \times N$ entries is represented by $\mathcal{Y} = \mathcal{Y}_{\text{obs}} \cup \mathcal{Y}_{\text{unobs}}$. A weight w_{mn}, $[m]_1^M, [n]_1^N$, is associated with each entry y_{mn} such that $w_{mn} = 1$ if $y_{mn} \in \mathcal{Y}_{\text{obs}}$ and $w_{mn} = 0$ for $y_{mn} \in \mathcal{Y}_{\text{unobs}}$. A weighted neighborhood structure $\mathcal{N}_{1m}$, is used to denote a set of rows that form the neighborhood of a row m along with the associated link strengths $\boldsymbol{\zeta}_{1m}$. Such neighborhoods can capture a variety of domain knowledge. For example, to represent must-link/cannot-link constraints [40], the neighborhood $\mathcal{N}_{1m}$ will consist of all the rows included in the must-link and cannot-link constraints involving the row m. For the must-link constraints, the link strengths can be set to a large positive value while for cannot-link constraints can be set to an equally large negative value. A similar weighted neighborhood $\mathcal{N}_{2n}$, $[n]_1^N$ is defined for each column n with link strengths ζ_{2n}.

The Bayesian Semi-supervised Co-clustering (BSCC henceforth) co-clusters a data matrix into KL co-clusters obtained as a cross-product of clustering the rows and columns into K and L clusters respectively. The cluster assignments for row m and column n are represented by $z_{1m} \in \{1, \ldots, K\}$ and $z_{2n} \in \{1, \ldots, L\}$ respectively. The neighborhood information is then incorporated in the form of separate Markov random field priors [29] over the set of cluster assignment variables $\mathcal{Z}_1$ and $\mathcal{Z}_2$. The joint prior distribution of the latent cluster assignment variables is then given by:

$$p(\mathcal{Z}_1 | \mathbb{N}_1, \boldsymbol{\zeta}_1) \;\propto\; \prod_m \exp\left(\sum_{i \in \mathcal{N}_{1m}} \zeta_{1mi} \mathbb{1}_{\{z_{1m} = z_{1i}\}} \right) \tag{168}$$

$$p(\mathcal{Z}_2 | \mathbb{N}_2, \zeta_2) \;\propto\; \prod_n \exp\left(\sum_{j \in \mathcal{N}_{2n}} \zeta_{2nj} \mathbb{1}_{\{z_{2n} = z_{2j}\}} \right) \tag{169}$$

The cluster assignments for a row-column pair, (z_{1m}, z_{2n}) together determine a co-cluster which then selects an exponential family distribution, $p_{\psi_{\mathcal{Y}}}(y_{mn} | \theta_{z_{11} z_{2n}})$ (out of KL such distributions), to generate a matrix entry y_{mn}. The parameters $\theta_{z_{11} z_{2n}}$ of the distribution are specific to the co-cluster (z_{1m}, z_{2n}). The overall joint distribution of the observed and latent variables is then given by:

$$p(\mathcal{Y}, \mathcal{Z}_1, \mathcal{Z}_2 | \boldsymbol{\Theta}, \mathbb{N}_1, \mathbb{N}_2, \boldsymbol{\zeta}_1, \zeta_2) = p(\mathcal{Z}_1 | \mathbb{N}_1, \boldsymbol{\zeta}_1) p(\mathcal{Z}_2 | \mathbb{N}_2, \zeta_2) \left(\prod_{m,n} p_{\psi_{\mathcal{Y}}}(y_{mn} | \theta_{z_{11} z_{2n}}) \right)$$

The free model parameters $\mathbf{\Theta}$ can be estimated by maximizing the incomplete log-likelihood via Expectation Maximization [28]. Computation of incomplete log-likelihood requires marginalization over all possible states of cluster assignments $\mathcal{Z}_1$ and $\mathcal{Z}_2$. Due to the correlations induced by the MRF priors, this marginalization requires a computation that is exponential in the size of the largest clique in the given neighborhood structures. To overcome this problem, we employ a fully factorized mean field approximation for an approximate inference.

As before, the true posterior distribution over the unobserved variables is approximated by the following parameterized distributions

$$q(\mathcal{Y}_{\text{unobs}}, \mathcal{Z}_1, \mathcal{Z}_2 | \phi_{mn}, \boldsymbol{r}_1, \boldsymbol{r}_2) = \left(\prod_{\substack{m,n \\ y_{mn} \in \mathcal{Y}_{\text{unobs}}}} q_{\psi_y}(y_{mn}|\phi_{mn}) \right) \left(\prod_m q(z_{1m}|\boldsymbol{r}_{1m}) \right) \left(\prod_n q(z_{2n}|\boldsymbol{r}_{2n}) \right) \tag{170}$$

where similar to the case of NA-BAE, $q_{\psi_y}(y_{mn}|\phi_{mn})$ is an exponential family distribution of the same form as the one assumed for the matrix entries and with natural parameter ϕ_{mn}. Variational distributions over cluster assignments $q(z_{1m}\boldsymbol{r}_{1m})$ and $q(z_{2n}|\boldsymbol{r}_{2n})$ follow discrete distributions over K and L clusters with parameters $\boldsymbol{r}_{1m}$, $\boldsymbol{r}_{2n}$ respectively. Following analysis of section 3.1, a variational mean field approximation for posterior inference then results in following updates for the variational parameters

$$\phi_{mn} = \sum_{k=1}^{K} \sum_{l=1}^{L} r_{1mk} r_{2nl} \theta_{kl} \tag{171}$$

$$r_{1mk} \propto \exp \left(\sum_{i \in \mathcal{N}_{1m}} \zeta_{1mi} r_{1ik} + \sum_{n=1}^{N} \sum_{l=1}^{L} r_{2nl} \left(w_{mn} \log p_{\psi_y}(y_{mn}|\theta_{z_{11} z_{2n}}) + (1 - w_{mn}) \mathbb{E}_q[\log p_{\psi_y}(y_{mn}|\theta_{z_{11} z_{2n}})] \right) \right) \tag{172}$$

$$r_{2nl} \propto \exp \left(\sum_{j \in \mathcal{N}_{2n}} \zeta_{2nj} r_{2nj} + \sum_{m=1}^{M} \sum_{k=1}^{K} r_{1mk} \left(w_{mn} \log p_{\psi_y}(y_{mn}|\theta_{z_{11} z_{2n}}) + (1 - w_{mn}) \mathbb{E}_q[\log p_{\psi_y}(y_{mn}|\theta_{z_{11} z_{2n}})] \right) \right) \tag{173}$$

where the expectation $\mathbb{E}_q[\log p_{\psi_y}(y_{mn}|\theta_{z_{11} z_{2n}})]$ is taken with respect to the variational distribution $q_{\psi_y}(y_{mn}|\phi_{mn})$ over the missing matrix entries. The coupled mean field equations can be iteratively satisfied for an approximate posterior inference which can then be used to construct a lower bound on the observed log-likelihood similar to (4). The lower bound can then be maximized with respect to the free model parameters to update the natural parameters $\mathbf{\Theta}$ of the exponential family distributions over the matrix entries. The update is given as follows:

$$\theta_{kl} = \nabla \psi_y^{-1} \left(\frac{\sum_{m,n=1}^{M,N} r_{1mk} r_{2nl} \left(w_{mn} y_{mn} + (1 - w_{mn}) \nabla_{\psi_y}(\phi_{mn}) \right)}{\sum_{m,n=1}^{M,N} r_{1mk} r_{2nl}} \right) \tag{174}$$

Note that the missing entries (represented by $w_{mn} = 0$) are replaced by their expected value, $\nabla_{\psi_y}(\phi_{mn})$ under their variational distribution. An EM style algorithm can then be derived wherein E-step variational posterior inference is done by updating the mean field equations to construct a tight lower bound on the observed log-likelihood. The optimized lower bound is then maximized with respect to free model parameters $\mathbf{\Theta}$, in the subsequent M-step to get an improved estimate of their values. Starting with an initial guess of $\mathbf{\Theta}$, the algorithm iterates between two steps until convergence.

12 Concluding Remarks

Side information aware Bayesian affinity estimation is a promising framework that efficiently incorporates multiple sources of side information including past affinities, entity attributes, temporal information, and/or neighborhood structures, within a Bayesian framework for an affinity estimation task. The use of exponential family distributions for modeling entity attributes as well as the affinity relationships renders great flexibility for modeling diverse data types in numerous domains. Embedding a factorized representation within the SABAE framework allows models with a strong generalization capability without losing the interpretability of a mixture model. Bayesian framework further allows an efficient modeling of the self-recording behavior of the affinity relationships leading to an improved generalization ability. Many additional useful tasks such as estimating missing entity attributes, efficient feature and model selection, a supervised ranking framework as well as a model for semi-supervised constrained co-clustering are obtained as side products of the SABAE framework.

While in this paper we have followed a parametric approach towards Bayesian modeling that requires the number of clusters as an input to the framework, the framework can easily be extended to non-parametric models by replacing the Dirichlet distribution priors with the corresponding process prior. This will enable an automatic estimation of the required number of clusters.

References

[1] D. Agarwal and S. Merugu. Predictive discrete latent factor models for large scale dyadic data. In *Proc. SIGKDD 2007.*

[2] D. Lee and H.Seung. Algorithms for non-negative matrix factorization. In *Proc. NIPS, 2001.*

[3] R. Salakhutdinov and A.Mnih. Probabilistic matrix factorization. In *Proc. NIPS, 2007*, pages 1257–1264, 2007.

[4] A. Banerjee, I.Dhillon, J. Ghosh and D. Modha. A generalized maximum entropy approach to Bregman co-clustering and matrix approximation. *JMLR*, volume 8, pages 1919–1986, 2007.

[5] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proc. SIGKDD 2008*, pages 426–434, 2008.

[6] Y.Koren. Collaborative filtering with temporal dynamics. In *Proc. SIGKDD 2009*, pages 447–456, 2009.

[7] R.Little and D.Rubin. Statistical analysis with missing data. John Wiley & Sons, Inc., 1987.

[8] D. Agarwal and B.C. Chen. Regression based latent factor models. In *Proc. SIGKDD 2009*, pages 19–28, 2009.

[9] D.M. Blei and J. Lafferty. Dynamic topic models. In *Proc. ICML 2006*, pages 113–120, 2006.

[10] T. Hofmann and J. Basilico. A joint framework for collaborative and content-based filtering. In *Proc. SIGIR 2004*, pages 550–551, 2204.

[11] Ilya Sutskever and Ruslan Salakhutdinov and Joshua Tenenbaum. Modeling relational data using bayesian clustered tensor factorization. In *Proc. NIPS 2009*, pages 1821–1828, 2009.

[12] A. Gunawardana and C. Meek. Tied boltzmann machines for cold start recommendations. In *Proc. ACM conference on Recommender Systems, 2008*, pages 19–26, 2008.

[13] M. Deodhar and J. Ghosh. Simultaneous co-clustering and learning from complex data. In *Proc. SIGKDD, 2007*, pages 250–259, 2007.

[14] Z. Lu, D. Aggarwal and I. Dhillon. A spatio-temporal approach to collaborative filtering. In *Proc. ACM conference on Recommender Systems, 2009*. pages 13–20, 2009.

[15] H. Ma, H. Yang, M. Lyu and I.King. Sorec: social recommendation using probabilistic matrix factorization. In *Proc. CIKM, 2008*, pages 931–940, 2008.

[16] S. Roweis, B. Marlin, R.Zemel and M. Slaney. Collaborative filtering and the missing at random assumption. In *Proc. Uncertainty in Artificial Intelligence, 2007*.

[17] B. Marlin, S. Roweis and R. Zemel. Unsupervised learning with non-ignorable missing data. In *Proc. AISTATS, 2005*, pages 222–229, 2005.

[18] A. Paterek. Improving regularized singular value decomposition for collaborative filtering. In *Proc. KDD Cup and Workshop, 2007*.

[19] E. Airoldi, D. M. Blei, S. E. Fienberg and E.P. Xing. Mixed membership stochastic blockmodels. In *JMLR*, volume 9, pages 1981–2014, 2008.

[20] Y.J. Lim and Y.W. Teh. Variational bayesian approach to movie rating prediction. In *Proc. KDD Cup and Workshop, 2007*.

[21] R. Salakhutdinov and A. Mnih. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In *Proc. ICML, 2008*, pages 880–887, 2008.

[22] D.M. Blei, A. Y. Ng and M. I. Jordan. Latent dirichlet allocation. In *JMLR*, volume 3, pages 993–1022, 2003.

[23] D. Agarwal and B.C. Chen. fLDA: matrix factorization through latent dirichlet allocation. In *Proc. ACM international conference on Web search and data mining, 2010*, pages 91–100, 2010.

[24] H. Shan and A. Banerjee. Bayesian co-clustering. In *Proc ICDM, 2008*, pages 530–539, 2008.

[25] P. Wang, C. Domeniconi and K.B. Laskey. Latent dirichlet bayesian co-clustering. In *Proc. ECML PKDD, 2009*, pages 522–537, 2009.

[26] H. Shan and A. Banerjee. Residual bayesian co-clustering and matrix approximation. In *Proc. SDM 2010*, pages 223–234, 2010.

[27] P. McCullagh and L.A. Nelder. Generalized linear models. Chapman and Hall, London, 1983.

[28] A. Dempster, N. Laird and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. In *Journal of Royal Statistical Society*, series B, volume 39(1), pages 1–38, 1977.

[29] M.J. Wainwright and M.I. Jordan. Graphical models, exponential families and variational inference. Now publishers Inc., 2008.

[30] R. Neal and G. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. In *Learning in Graphical Models*, pages 355–386, 1998.

[31] C. Bishop. Pattern recognition and machine learning. *Information science and statistics*, 2007.

[32] S. Kirkpatrick, C. Gelatt and M. Vecchi. Optimization by simulated annealing. *Science*, 220(4598): pages 671–680, 1983.

[33] R. M. Bell, Y. Koren and C. Volinsky. The bellkor solution to the Netflix prize.

[34] P.J. Lenk. The logistic normal distribution for bayesian, nonparametric, predictive densities. In *Journal of the Americal Statistical Association*, volume 83(402), pages 509–516, 1988.

[35] B. Chen, J. Paisley and L. Carin. Sparse linear regression with beta process priors. In *Proc ICASSP, 2010.*

[36] N. Friedman and Y. Singer. Efficient bayesian parameter estimation in large discrete domains. In *Proc. NIPS 1999.*

[37] C. Wang and D.M. Blei. Decoupling sparsity and smootheness in the discrete hierarchical dirichlet process. In *Proc. NIPS 2009.*

[38] J. Duchi, L. Mackey and M. Jordan. On consistency of ranking algorithms. In *Proc. ICML 2010.*

[39] D. Cossock and T. Zhang. Statistical analysis of Bayes optimal subset ranking. In *IEEE Transactions on Information Theory*, volume 16, pages 1274–1286, 2008.

[40] S. Basu, M. Bilenko, A. Banerjee, and R. Mooney. Probabilistic semi-supervised clustering with constraints. In *Semi-Supervised Learning*, MIT Press, 2006.

[41] R. Kalman. A new approach to linear filtering and prediction problems. In *Transactions of the AMSE - Journal of Basic Engineering*, volume 82, pages 35–45, 1960.

[42] D. Bertsekas. Nonlinear Programming. *Athena Scientific*, 1999.

A Variational Inference using Mean Field Approximation (MFA)

A maximum likelihood approach to parameter estimation generally involves maximization of the observed log-likelihood $\log p(\mathcal{X}|\mathbf{\Theta})$ with respect to the free model parameters, i.e.,

$$\begin{aligned}\mathbf{\Theta}^*_{ML} &= \arg\max_{\mathbf{\Theta}} \log p(\mathcal{X}|\mathbf{\Theta}) && \text{(A1)}\\ &= \arg\max_{\mathbf{\Theta}} \log \int_{\mathcal{Z}} p(\mathcal{X},\mathcal{Z}|\mathbf{\Theta})d\mathcal{Z} && \text{(A2)}\end{aligned}$$

where $\mathcal{X}$ and $\mathcal{Z}$ are sets of observed and hidden variables respectively. In the presence of hidden variables, the maximum likelihood estimate is often done using the Expectation-Maximization (EM) algorithm [28]. The following lemma forms the basis of the EM algorithm [29].

Lemma 1. *Let $\mathcal{X}$ denote a set of all the observed variables and $\mathcal{Z}$ a set of the hidden variables in a Bayesian network. Then, the observed log-likelihood can be lower bounded as follows*

$$\log p(\mathcal{X},\mathcal{Z}|\mathbf{\Theta}) \geq \mathcal{F}(Q,\mathbf{\Theta})$$

where

$$\mathcal{F}(Q,\mathbf{\Theta}) = -\int_{\mathcal{Z}} Q(\mathcal{Z})\log Q(\mathcal{Z})d\mathcal{Z} + \int_{\mathcal{Z}} Q(\mathcal{Z})\log p(\mathcal{X},\mathcal{Z}|\mathbf{\Theta})d\mathcal{Z} \qquad \text{(A3)}$$

for some distribution Q and the free model parameters $\mathbf{\Theta}$.

Proof. The proof follows from the Jensen's inequality and the concavity of the log function.

$$\begin{aligned}\log p(\mathcal{X}|\mathbf{\Theta}) &= \log \int_{\mathcal{Z}} \frac{Q(\mathcal{Z})}{Q(\mathcal{Z})} p(\mathcal{X},\mathcal{Z}|\mathbf{\Theta})d\mathcal{Z}\\ &\geq \int_{\mathcal{Z}} Q(\mathcal{Z})\log \frac{p(\mathcal{X},\mathcal{Z}|\mathbf{\Theta})}{Q(\mathcal{Z})}d\mathcal{Z}\\ &= -\int_{\mathcal{Z}} Q(\mathcal{Z})\log Q(\mathcal{Z})d\mathcal{Z} + \int_{\mathcal{Z}} Q(\mathcal{Z})\log p(\mathcal{X},\mathcal{Z}|\mathbf{\Theta})d\mathcal{Z}\\ &= \mathcal{F}(Q,\mathbf{\Theta})\end{aligned}$$

□

Starting from an initial estimate of the parameters, $\mathbf{\Theta}_0$, the EM algorithm alternates between maximizing the lower bound $\mathcal{F}$ with respect to Q (E-step) and $\mathbf{\Theta}$ (M-step), respectively, holding the other fixed. The following lemma shows that maximization the lower bound with respect to the distribution Q in the E-step makes the bound exact, so that the M-step is guranteed to increase the observed log-likelihood with respect to the parameters.

Lemma 2. *Let $\mathcal{F}(Q,\mathbf{\Theta})$ denote a lower bound on the observed log-likelihood of the form in* (A3)*, then*

$$Q^* = p(\mathcal{Z}|\mathcal{X},\mathbf{\Theta}) = \arg\max_{Q} \mathcal{F}(Q,\mathbf{\Theta})$$

and $\mathcal{F}(Q^*,\mathbf{\Theta}) = \log p(\mathcal{X}|\mathbf{\Theta})$.

Proof. The lower bound on the observed log-likelihood is

$$\begin{aligned}
\mathcal{F}(Q, \mathbf{\Theta}) &= -\int_{\mathcal{Z}} Q(\mathcal{Z}) \log Q(\mathcal{Z}) d\mathcal{Z} + \int_{\mathcal{Z}} Q(\mathcal{Z}) \log p(\mathcal{X}, \mathcal{Z}|\mathbf{\Theta}) d\mathcal{Z} \\
&= -\int_{\mathcal{Z}} Q(\mathcal{Z}) \log \frac{Q(\mathcal{Z})}{p(\mathcal{Z}|\mathcal{X}, \mathbf{\Theta})} d\mathcal{Z} + \int_{\mathcal{Z}} Q(\mathcal{Z}) \log \frac{p(\mathcal{X}, \mathcal{Z}|\mathbf{\Theta})}{p(\mathcal{Z}|\mathcal{X}, \mathbf{\Theta})} d\mathcal{Z} \\
&= \log p(\mathcal{X}|\mathbf{\Theta}) - \mathrm{KL}(Q \parallel p(\mathcal{Z}|\mathcal{X}, \mathbf{\Theta}))
\end{aligned}$$

Maximum is attained when the KL-divergence $\mathrm{KL}(Q \parallel p(\mathcal{Z}|\mathcal{X}, \mathbf{\Theta}))$ is zero, which is uniquely achieved for $Q^* = p(\mathcal{Z}|\mathcal{X}, \mathbf{\Theta})$ at which point the bound becomes an equality for $\log p(\mathcal{X}|\mathbf{\Theta})$. □

However, in many cases, computation of the true posterior distribution, $p(\mathcal{Z}|\mathcal{X}, \mathbf{\Theta})$ is intractable. To overcome this problem, the distribution Q is restricted to a certain family of distributions. The optimal distribution within this restricted class is then obtained by minimizing the KL-divergence to the true posterior distribution. The approximating distribution is known as a *variational distribution* [29].

There are a number of ways in which the family of possible distributions can be restricted. One way of restricting the approximating distributions is to use a parameteric distribution $Q(\mathcal{Z}|\mathbf{\Phi})$ determined by a set of parameters $\mathbf{\Phi}$, known as *variational parameters*. In the E-step, the lower bound then becomes a function of variational parameters, and standard non-linear optimization methods can be employed to obtain the optimal values of these parameters. Yet another way to restrict the family of approximationg distributions is to assume a certain conditional independence structure over the hidden variables $\mathcal{Z}$. For example, one can assume a family of fully factorized distributions of the following form

$$Q = \prod_i q_i(z_i) \tag{A4}$$

This fully factorized assumption is often known as a *mean field approximation* in statistical mechanics. The following lemma derieves the expression for optimal variational distribution subject to a full factorization assumption.

Lemma 3. *Let* $\mathcal{Q} = \{Q\}$ *be a family of factorized distributions of the form in* (A4)*. Then the optimal factorized distribution corresponding to the tightest lower bound is given by,*

$$Q^* = \prod_i q_i^*(z_i) = \arg\max_{Q \in \mathcal{Q}} \mathcal{F}(Q, \mathbf{\Theta}) \quad \textit{such that} \quad q_i^*(z_i) \propto \exp\left(\mathbb{E}_{-i}[\log p(\mathcal{X}, \mathcal{Z}|\mathbf{\Theta})]\right)$$

where $\mathbb{E}_{-i}[\log p(\mathcal{X}, \mathcal{Z}|\mathbf{\Theta})]$ *denotes a conditional expectation conditioned on* z_i*.*

Proof. Using lemma 2, the optimal distribution $Q \in \mathcal{Q}$ is given by

$$Q^* = \arg\min_{Q \in \mathcal{Q}} \quad \mathrm{KL}(Q \parallel p(\mathcal{Z}|\mathcal{X}, \mathbf{\Theta}))$$

where the KL-divergence can be expressed as

$$\begin{aligned}
\mathrm{KL}(Q \parallel p(\mathcal{Z}|\mathcal{X}, \mathbf{\Theta})) &= \sum_i \int_{z_i} q_i(z_i) \log q_i(z_i) dz_i - \int_{z_i} q_i(z_i) \left\{ \int_{\mathcal{Z}_{-i}} \log p(\mathcal{Z}|\mathcal{X}, \mathbf{\Theta}) \prod_{j \neq i} q_j(z_j) d\mathcal{Z}_{-i} \right\} dz_i \\
&= \sum_{j \neq i} \int_{z_j} q_j(z_j) \log q_j(z_j) dz_j + \int_{z_i} q_i(z_i) \log \frac{q_i(z_i)}{\exp\left(\mathbb{E}_{-i}[\log p(\mathcal{X}, \mathcal{Z}|\mathbf{\Theta})\right)]} dz_i
\end{aligned}$$

The second term in the above expression is a KL-divergence. Keeping $\{q_{j \neq i}(z_j)\}$ fixed, the optimum with respect to $q_i(z_i)$ is attained when KL-divergence is zero, i.e. $q_i^*(z_i) \propto \exp\left(\mathbb{E}_{-i}[\log p(\mathcal{X}, \mathcal{Z}|\mathbf{\Theta})]\right)$. □

The above lemma shows that the optimal variational distribution subject to the factorization constraint is given by a set of consistency conditions over different factors of the hidden variables. These coupled equations are known as *mean field equations* and can be satisfied iteratively. Convergence is guaranteed because the bound $\mathcal{F}$ is convex with respect to each of the factors [42].

B MFA for Bayesian Affinity Estimation

This appendix illustrates the derivation of a MFA based expectation maximization algorithm for parameter estimation of a Latent Dirichlet Attribute Aware Bayesian Affinity Estimation framework (LD-AA-BAE). The techniques introduced in this appendix are also used for derieving updates for rest of the models in the paper and the same analysis can be easily extended. For the purpose of exposition, we however, concentrate only on the LD-AA-BAE model.

The joint distribution over all observable and latent variables for the LD-AA-BAE model is given by:

$$p(\mathcal{Y}, \mathcal{X}_1, \mathcal{X}_2, \mathcal{Z}_1, \mathcal{Z}_2, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2 | \alpha_1, \alpha_2, \boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2, \boldsymbol{\beta}) =$$

$$p(\boldsymbol{\pi}_1|\alpha_1)p(\boldsymbol{\pi}_2|\alpha_2)\left(\prod_m p(z_{1m}|\boldsymbol{\pi}_1)p_{\psi_1}(\boldsymbol{x}_{1m}|\boldsymbol{\theta}_{1z_{1m}})\right)\left(\prod_n p(z_{2n}|\boldsymbol{\pi}_2)p_{\psi_2}(\boldsymbol{x}_{2n}|\boldsymbol{\theta}_{2z_{2n}})\right)\left(\prod_{m,n} p_{\psi_\mathcal{Y}}(y_{mn}|\boldsymbol{\beta}^\dagger_{z_{1m}z_{2n}}\boldsymbol{x}_{mn})\right) \tag{B1}$$

The approximate variational distribution Q over the hidden variables is

$$Q(\mathcal{Y}_{\text{unobs}}, \mathcal{Z}_1, \mathcal{Z}_2, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2) = q(\boldsymbol{\pi}_1|\gamma_1)q(\boldsymbol{\pi}_{2|}\gamma_2)\left(\prod_{\substack{m,n \\ y_{mn}\in\mathcal{Y}_{\text{unobs}}}} q(y_{mn}|\phi_{mn})\right)\left(\prod_m q(z_{1m}|r_{1m})\right)\left(\prod_n q(z_{2n}|r_{2n})\right) \tag{B2}$$

The updates for factors corresponding to the optimal variational distribution is obtained using lemma 3. **E-step Update for** $q^*(y_{mn}|\phi_{mn})$: Collecting terms containing the affinities y_{mn} in the conditional expectation of the complete log-likelihood, we obtain

$$q^*(y_{mn}) \propto p_0(y_{mn}) \exp\left(\sum_{K,L=1}^{K,L} r_{1mk}r_{2nl}\langle y_{mn}, \boldsymbol{\beta}^\dagger_{kl}\boldsymbol{x}_{mn}\rangle\right)$$

which shows that variational distribution for the missing affinities is an exponential family distribution having the same form as the one assumed for the affinities with the natural parameter given by:

$$\phi_{mn} = \sum_{k,l=1}^{K,L} r_{1mk}r_{2nl}\left(\boldsymbol{\beta}^\dagger_{kl}\boldsymbol{x}_{mn}\right) \tag{B3}$$

E-step Updates for $q^*(\boldsymbol{\pi}_1|\gamma_1)$ and $q^*(\boldsymbol{\pi}_{2|}\gamma_2)$: Conditional expectation with respect to the mixing coefficients $\boldsymbol{\pi}_1$ yields,

$$\begin{aligned} q^*(\boldsymbol{\pi}_1) &\propto \exp\left(\sum_{k=1}^{K}\left(\alpha_{1k} + \sum_{m=1}^{M} r_{1mk}\right)\log \pi_{1k}\right) \\ &= \prod_{k=1}^{K} (\pi_{1k})^{\left(\alpha_{1k}+\sum_{m=1}^{M} r_{1mk}\right)} \end{aligned}$$

Easy to see that, the optimal variational distribution $q^*(\boldsymbol{\pi}_1|\boldsymbol{\gamma}_1)$ is a Dirichlet distribution over a K-simplex with parameters given by:

$$\gamma_{1k} = \alpha_{1k} + \sum_{m=1}^{M} r_{1mk} \tag{B4}$$

Similarly, $q^*(\boldsymbol{\pi}_2|\boldsymbol{\gamma}_2)$ is a Dirichlet distribution over a L-simplex with parameters:

$$\gamma_{2l} = \alpha_{2l} + \sum_{n=1}^{N} r_{2nl} \tag{B5}$$

E-step Updates for $q(z_{1m}|r_{1m})$ and $q(z_{2n}|r_{2n})$: Conditional expectation with respect to discrete cluster assignment variable z_{1mk} for the cluster k results in the following update:

$$q^*(z_{1mk} = 1) = r_{1mk} \propto \exp\left(\log p_{\psi_1}(\boldsymbol{x}_{1m}|\boldsymbol{\theta}_{1k}) + \Psi(\gamma_{1k}) - \Psi\left(\sum_{k'=1}^{K} \gamma_{1k'}\right) + \right.$$

$$\left. \sum_{n=1}^{N} \sum_{l=1}^{L} r_{2nl} \left(w_{mn} \log p_{\psi_y}(y_{mn}|\boldsymbol{\beta}_{kl}^{\dagger}\boldsymbol{x}_{mn}) + (1 - w_{mn})\mathbb{E}_q\left[\log p_{\psi_y}(y_{mn}|\boldsymbol{\beta}_{kl}^{\dagger}\boldsymbol{x}_{mn})\right]\right)\right) \tag{B6}$$

The first term is the log-likelihood of the entity attributes, the second term is the expectation of $\log \pi_{1k}$ with respect to the variational Dirichlet distribution while the last term involves the log-likelihood of all the affinities associated with the entity e_{1m}. The known log-likelihood is used if the affinity is observed ($w_{mn} = 0$), while the log-likelihood for the missing affinities is replaced by the corresponding expecations under the variational distribution $q^*(y_{mn}|\phi_{mn})$. Analogously, the update equation for the cluster assignment variable $q^*(z_{2nl} = 1)$ is given by:

$$q^*(z_{2nl} = 1) = r_{2nl} \propto \exp\left(\log p_{\psi_2}(\boldsymbol{x}_{2n}|\boldsymbol{\theta}_{2l}) + \Psi(\gamma_{2l}) - \Psi\left(\sum_{l'=1}^{L} \gamma_{2l'}\right) + \right.$$

$$\left. \sum_{m=1}^{M} \sum_{k=1}^{K} r_{1mk} \left(w_{mn} \log p_{\psi_y}(y_{mn}|\boldsymbol{\beta}_{kl}^{\dagger}\boldsymbol{x}_{mn}) + (1 - w_{mn})\mathbb{E}_q\left[\log p_{\psi_y}(y_{mn}|\boldsymbol{\beta}_{kl}^{\dagger}\boldsymbol{x}_{mn})\right]\right)\right) \tag{B7}$$

M-step Updates for $\boldsymbol{\theta}_{1k}$ and $\boldsymbol{\theta}_{2l}$: Taking expectation of the complete log-likelihood with respect to the variational distribution, we obtain the following expression for the lower bound $\mathcal{F}$ as a function of the entity attributes parameters:

$$\mathcal{F}(\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2) = \sum_{m=1}^{M} \sum_{k=1}^{K} r_{1mk} \log p_{\psi_1}(\boldsymbol{x}_{1m}|\boldsymbol{\theta}_{1k}) + \sum_{n=1}^{N} \sum_{l=1}^{L} r_{2nl} \log p_{\psi_2}(\boldsymbol{x}_{2n}|\boldsymbol{\theta}_{2l})$$

Taking partial derivatives with respect to $\boldsymbol{\theta}_{1k}$ and $\boldsymbol{\theta}_{2l}$, we obtain the following updates:

$$\boldsymbol{\theta}_{1k} = \nabla\psi_1^{-1}\left(\frac{\sum_{m=1}^{M} r_{1mk}\boldsymbol{x}_{1m}}{\sum_{m=1}^{M} r_{1mk}}\right) \tag{B8}$$

$$\boldsymbol{\theta}_{2l} = \nabla\psi_2^{-1}\left(\frac{\sum_{n=1}^{N} r_{2nl}\boldsymbol{x}_{2n}}{\sum_{n=1}^{N} r_{2nl}}\right) \tag{B9}$$

M-step Updates for $\boldsymbol{\beta}_{kl}$: Collecting terms containing the GLM coefficients in the lower bound, we obtain:

$$\mathcal{F}(\boldsymbol{\beta}_{kl}) = \sum_{m=1}^{M}\sum_{n=1}^{N} r_{1mk}r_{2nl}\left[\left\langle (w_{mn}y_{mn} + (1-w_{mn})\nabla\psi_{\mathcal{Y}}(\phi_{mn})), \boldsymbol{\beta}^{\dagger}\boldsymbol{x}_{mn}\right\rangle - \psi_{\mathcal{Y}}\left(\boldsymbol{\beta}^{\dagger}\boldsymbol{x}_{mn}\right)\right]$$

As earlier, the missing affinities are replaced by corresponding expected values under the variational exponential family distribution. The lower bound can be maximized using a gradient ascent method. The expressions for the gradient and the gradient-ascent updates are obtained as follows:

$$\nabla\mathcal{F}(\boldsymbol{\beta}_{kl}) = \sum_{m=1}^{M}\sum_{n=1}^{N} r_{1mk}r_{2nl}\left[(w_{mn}y_{mn} + (1-w_{mn})\nabla\psi_{\mathcal{Y}}(\phi_{mn})) - \nabla\psi_{\mathcal{Y}}\left(\boldsymbol{\beta}^{\dagger}\boldsymbol{x}_{mn}\right)\right]\boldsymbol{x}_{mn} \qquad \text{(B10)}$$

$$\boldsymbol{\beta}_{kl}^{t+1} = \boldsymbol{\beta}_{kl}^{t} + \eta\nabla\mathcal{F}(\boldsymbol{\beta}_{kl}) \qquad \text{(B11)}$$

where η is the step-size for the update.

M-step Updates for $\boldsymbol{\alpha}_1$ and $\boldsymbol{\alpha}_2$: The expression for the lower bound as a function of the Dirichlet parameters $\boldsymbol{\alpha}_1$ is:

$$\mathcal{F}(\boldsymbol{\alpha}_1) = \log\frac{\Gamma(\sum_{k=1}^{K}\alpha_{1k})}{\prod_{k=1}^{K}\Gamma(\alpha_{1k})} + \sum_{k=1}^{K}\left(\alpha_{1k} + \sum_{m=1}^{M} r_{1mk} - 1\right)\left(\Psi(\gamma_{1k}) - \Psi\left(\sum_{k'=1}^{K}\gamma_{1k'}\right)\right)$$

Taking derivative with respect to α_{1k} yield:

$$\frac{\partial\mathcal{F}}{\partial\alpha_{1k}} = \Psi\left(\sum_{k'=1}^{K}\alpha_{1k}\right) - \Psi(\alpha_{1k}) + \Psi\left(\sum_{k'=1}^{K}\gamma_{1k}\right) - \Psi(\gamma_{1k})$$

Note that the update for α_{1k} depends on $\{\alpha_{1k'}, [k']_1^K, k' \neq k\}$, so a closed form solution cannot be obtained. Following [22], Newton-Raphson's method can then be used to update the parameters. The Hessian H is given by

$$\begin{aligned} H(k,k) &= \frac{\partial^2\mathcal{F}}{\partial\alpha_{1k}^2} = \Psi'\left(\sum_{k'=1}^{K}\alpha_{1k}\right) - \Psi'(\alpha_{1k}) \\ H(k,k') &= \frac{\partial^2\mathcal{F}}{\partial\alpha_{1k}\partial\alpha_{1k'}} = \Psi'\left(\sum_{k'=1}^{K}\alpha_{1k}\right) \quad (k' \neq k) \end{aligned}$$

The update can then be obtained as follows:

$$\boldsymbol{\alpha}_1^{t+1} = \boldsymbol{\alpha}_1^{t} + \eta H^{-1}\nabla(\boldsymbol{\alpha}_1) \qquad \text{(B12)}$$

The step-size η can be adapted to satisfy the positivity constraint for the Dirichlet parameters. Similar method is followed for update of $\boldsymbol{\alpha}_2$.

C Variational Kalman Filtering

This appendix derieves a Kalman filtering based method for updating the variational distributions over the time varying latent variables for Bayesian Affinity Estimation with Temporal Dynamics. The state space model for these latent variables is formulated as follows:

$$\begin{aligned}
\alpha_{1,t}|\alpha_{1,t-1} &\sim N(\alpha_{1,t}|\alpha_{1,t-1}, \delta_1^2 \boldsymbol{I}) \\
\alpha_{2,t}|\alpha_{2,t-1} &\sim N(\alpha_{2,t}|\alpha_{2,t-1}, \delta_2^2 \boldsymbol{I}) \\
\beta_{kl,t}|\beta_{kl,t-1} &\sim N(\beta_{kl,t}|\beta_{kl,t-1}, \omega^2 \boldsymbol{I})
\end{aligned}$$

To facilitate the use of Kalman filtering in updating the linear state space model, the variational parameters for the time varying latent variables are assumed to be *Gaussian observations* of the filter:

$$\begin{aligned}
\hat{\alpha}_{1,t}|\alpha_{1,t} &\sim N(\hat{\alpha}_{1,t}|\alpha_{1,t}, \hat{v}^2_{\alpha_1,t} \boldsymbol{I}) \\
\hat{\alpha}_{2,t}|\alpha_{2,t} &\sim N(\hat{\alpha}_{2,t}|\alpha_{2,t}, \hat{v}^2_{\alpha_2,t} \boldsymbol{I}) \\
\hat{\beta}_{kl,t}|\beta_{kl,t} &\sim N(\hat{\beta}_{kl,t}|\beta_{kl,t}, \hat{v}^2_{\beta_{kl},t} \boldsymbol{I})
\end{aligned}$$

We follow the analysis in [9] for updating the dynamic variational parameters using Kalman filtering.
Calculation of $\tilde{m}_{\alpha_1,t}$, $\tilde{V}_{\alpha_1,t}$, $\tilde{m}_{\alpha_2,t}$, $\tilde{V}_{\alpha_2,t}$, $\tilde{m}_{\beta_{kl},t}$, $\tilde{V}_{\beta_{kl},t}$:
The smoothed estimators $\tilde{m}_{\alpha_1,t}, \tilde{V}_{\alpha_1,t}, \tilde{m}_{\alpha_2,t}, \tilde{V}_{\alpha_2,t}, \tilde{m}_{\beta_{kl},t}$ and $\tilde{V}_{\beta_{kl},t}$ are derived by the standard Kalman backward recursion.

$$\begin{aligned}
\tilde{m}_{\alpha_1,t-1} &= \left(\frac{\delta_1^2}{V_{\alpha_1,t} + \delta_1^2}\right) m_{\alpha_1,t-1} + \left(1 - \frac{\delta_1^2}{V_{\alpha_1,t} + \delta_1^2}\right) \tilde{m}_{\alpha_1,t-1} \\
\tilde{m}_{\alpha_2,t-1} &= \left(\frac{\delta_2^2}{V_{\alpha_2,t} + \delta_2^2}\right) m_{\alpha_2,t-1} + \left(1 - \frac{\delta_2^2}{V_{\alpha_2,t} + \delta_2^2}\right) \tilde{m}_{\alpha_2,t-1} \\
\tilde{m}_{\beta_{kl},t-1} &= \left(\frac{\omega^2}{V_{\beta_{kl},t} + \omega^2}\right) m_{\beta_{kl},t-1} + \left(1 - \frac{\omega^2}{V_{\beta_{kl},t} + \omega^2}\right) \tilde{m}_{\beta_{kl},t-1} \\
\tilde{V}_{\alpha_1,t-1} &= V_{\alpha_1,t-1} + \left(\frac{V_{\alpha_1,t-1}}{V_{\alpha_1,t-1} + \delta_1^2}\right)^2 \left(\tilde{V}_{\alpha_1,t} - (V_{\alpha_1,t-1} + \delta_1^2)\right) \\
\tilde{V}_{\alpha_2,t-1} &= V_{\alpha_2,t-1} + \left(\frac{V_{\alpha_2,t-1}}{V_{\alpha_2,t-1} + \delta_2^2}\right)^2 \left(\tilde{V}_{\alpha_2,t} - (V_{\alpha_2,t-1} + \delta_2^2)\right) \\
\tilde{V}_{\beta_{kl},t-1} &= V_{\beta_{kl},t-1} + \left(\frac{V_{\beta_{kl},t-1}}{V_{\beta_{kl},t-1} + \omega^2}\right)^2 \left(\tilde{V}_{\beta_{kl},t} - (V_{\beta_{kl},t-1} + \omega^2)\right)
\end{aligned}$$

with initial conditions $\tilde{m}_{.,T} = m_{.,T}$ and $\tilde{V}_{.,T} = V_{.,T}$. The values of $m_{.,T}$ and $V_{.,T}$ are computed by the standard forward Kalman Filter equations as follows.
Calculation of $m_{\alpha_1,t}$, $V_{\alpha_1,t}$, $m_{\alpha_2,t}$, $V_{\alpha_2,t}$, $m_{\beta_{kl},t}$, $V_{\beta_{kl},t}$
$m_{\alpha_1,t}$, $V_{\alpha_1,t}$, $m_{\alpha_2,t}$, $V_{\alpha_2,t}$, $m_{\beta_{kl},t}$, $V_{\beta_{kl},t}$ are computed and stored for every time step t using forward Kalman

Filter updates. The Kalman recursion formulas are then given as follows:

$$m_{\alpha_1,t} = \left(\frac{\hat{v}^2_{\alpha_1,t}}{V_{\alpha_1,t-1} + \delta_1^2 + \hat{v}^2_{\alpha_1,t}}\right) m_{\alpha_1,t-1} + \left(1 - \frac{\hat{v}^2_{\alpha_1,t}}{V_{\alpha_1,t} + \delta_1^2 + \hat{v}^2_{\alpha_1,t}}\right)\hat{\alpha}_{1,t} \tag{C1}$$

$$m_{\alpha_2,t} = \left(\frac{\hat{v}^2_{\alpha_2,t}}{V_{\alpha_2,t-1} + \delta_2^2 + \hat{v}^2_{\alpha_2,t}}\right) m_{\alpha_2,t-1} + \left(1 - \frac{\hat{v}^2_{\alpha_2,t}}{V_{\alpha_2,t} + \delta_2^2 + \hat{v}^2_{\alpha_2,t}}\right)\hat{\alpha}_{2,t} \tag{C2}$$

$$m_{\beta_{kl},t} = \left(\frac{\hat{v}^2_{\beta_{kl},t}}{V_{\beta_{kl},t-1} + \omega^2 + \hat{v}^2_{\beta_{kl},t}}\right) m_{\beta_{kl},t-1} + \left(1 - \frac{\hat{v}^2_{\beta_{kl},t}}{V_{\beta_{kl},t} + \omega^2 + \hat{v}^2_{\beta_{kl},t}}\right)\hat{\beta}_{kl,t} \tag{C3}$$

Similarly the update for the variances is obtained by the following forward Kalman filtering equations:

$$V_{\alpha_1,t} = \left(\frac{\hat{v}^2_{\alpha_1,t}}{V_{\alpha_1,t-1} + \delta_1^2 + \hat{v}^2_{\alpha_1,t}}\right)\left(V_{\alpha_1,t-1} + \delta_1^2\right) \tag{C4}$$

$$V_{\alpha_2,t} = \left(\frac{\hat{v}^2_{\alpha_2,t}}{V_{\alpha_2,t-1} + \delta_2^2 + \hat{v}^2_{\alpha_2,t}}\right)\left(V_{\alpha_2,t-1} + \delta_2^2\right) \tag{C5}$$

$$V_{\beta_{kl},t} = \left(\frac{\hat{v}^2_{\beta_{kl},t}}{V_{\beta_{kl},t-1} + \omega^2 + \hat{v}^2_{\beta_{kl},t}}\right)\left(V_{\beta_{kl},t-1} + \omega^2\right) \tag{C6}$$

with initial conditions specified by fixed $m_{\cdot,0}$ and $V_{\cdot,0}$. Here we notice that these equations contain the variational observations, $\hat{\alpha}_1, \hat{\alpha}_2, \hat{\beta}_{kl,t}, \hat{v}^2_{\alpha_1,t}, \hat{v}^2_{\alpha_2,t}$ and $\hat{v}^2_{\beta_{kl},t}$. These are computed by maximizing the lower bound on the observed log-likelihood.

Calculation of Variational Observations: First, we introduce the standard backward recursions for $\partial\tilde{m}_{\alpha_1,t}/\partial\hat{\alpha}_{1,s}, \partial\tilde{m}_{\alpha_2,t}/\partial\hat{\alpha}_{2,s}$, and $\partial\tilde{m}_{\beta_{kl},t}/\partial\hat{\beta}_{kl,s}$.

$$\frac{\partial\tilde{m}_{\alpha_1,t-1}}{\hat{\alpha}_{1,s}} = \left(\frac{\delta_1^2}{V_{\alpha_1,t} + \delta_1^2}\right)\frac{\partial m_{\alpha_1,t-1}}{\hat{\alpha}_{1,s}} + \left(1 - \frac{\delta_1^2}{V_{\alpha_1,t} + \delta_1^2}\right)\frac{\partial\tilde{m}_{\alpha_1,t}}{\hat{\alpha}_{1,s}} \tag{C7}$$

$$\frac{\partial\tilde{m}_{\alpha_2,t-1}}{\hat{\alpha}_{2,s}} = \left(\frac{\delta_2^2}{V_{\alpha_2,t} + \delta_2^2}\right)\frac{\partial m_{\alpha_2,t-1}}{\hat{\alpha}_{2,s}} + \left(1 - \frac{\delta_2^2}{V_{\alpha_2,t} + \delta_2^2}\right)\frac{\partial\tilde{m}_{\alpha_2,t}}{\hat{\alpha}_{2,s}} \tag{C8}$$

$$\frac{\partial\tilde{m}_{\beta_{kl},t-1}}{\hat{\beta}_{kl,s}} = \left(\frac{\omega^2}{V_{\beta_{kl},t} + \omega^2}\right)\frac{\partial m_{\beta_{kl},t-1}}{\hat{\beta}_{kl,s}} + \left(1 - \frac{\omega^2}{V_{\beta_{kl},t} + \omega^2}\right)\frac{\partial\tilde{m}_{\beta_{kl},t}}{\hat{\beta}_{kl,s}} \tag{C9}$$

with initial conditions $\partial\tilde{m}_{\alpha_1,T}/\hat{\alpha}_{1,s} = \partial m_{\alpha_1,T}/\hat{\alpha}_{1,s}, \partial\tilde{m}_{\alpha_2,T}/\hat{\alpha}_{2,s} = \partial m_{\alpha_2,T}/\hat{\alpha}_{2,s}$ and $\partial\tilde{m}_{\beta_{kl},T}/\hat{\beta}_{kl,s} = \partial m_{\beta_{kl},T}/\hat{\beta}_{kl,s}$. Also, $\partial m_{\alpha_1,T}/\hat{\alpha}_{1,s}, \partial m_{\alpha_2,T}/\hat{\alpha}_{2,s}$ and $\partial m_{\beta_{kl},T}/\hat{\beta}_{kl,s}$ are derived by the following backward recursions.

$$\frac{\partial m_{\alpha_1,t}}{\partial\hat{\alpha}_{1,s}} = \left(\frac{\hat{v}^2_{\alpha_1,t}}{V_{\alpha_1,t-1} + \delta_1^2 + \hat{v}^2_{\alpha_1,t}}\right)\frac{\partial m_{\alpha_1,t-1}}{\partial\hat{\alpha}_{1,s}} + \left(1 - \frac{\hat{v}^2_{\alpha_1,t}}{V_{\alpha_1,t} + \delta_1^2 + \hat{v}^2_{\alpha_1,t}}\right)\delta_{s,t} \tag{C10}$$

$$\frac{\partial m_{\alpha_2,t}}{\partial\hat{\alpha}_{2,s}} = \left(\frac{\hat{v}^2_{\alpha_2,t}}{V_{\alpha_2,t-1} + \delta_2^2 + \hat{v}^2_{\alpha_2,t}}\right)\frac{\partial m_{\alpha_2,t-1}}{\partial\hat{\alpha}_{2,s}} + \left(1 - \frac{\hat{v}^2_{\alpha_2,t}}{V_{\alpha_2,t} + \delta_2^2 + \hat{v}^2_{\alpha_2,t}}\right)\delta_{s,t} \tag{C11}$$

$$\frac{\partial m_{\beta_{kl},t}}{\partial\hat{\beta}_{kl,s}} = \left(\frac{\hat{v}^2_{\beta_{kl},t}}{V_{\beta_{kl},t-1} + \omega^2 + \hat{v}^2_{\beta_{kl},t}}\right)\frac{\partial m_{\beta_{kl},t-1}}{\partial\hat{\beta}_{kl,s}} + \left(1 - \frac{\hat{v}^2_{\beta_{kl},t}}{V_{\beta_{kl},t} + \omega^2 + \hat{v}^2_{\beta_{kl},t}}\right)\delta_{s,t} \tag{C12}$$

with initial conditions $\partial m_{\alpha_1,0}/\hat{\alpha}_{1,s} = 0, \partial m_{\alpha_2,0}/\hat{\alpha}_{2,s} = 0$ and $\partial m_{\beta_{kl},T}/\hat{\beta}_{kl,s} = 0$ where $\delta_{s,t}$ denotes Kronecker delta. Similar backward recursions can be formulated for $\partial\tilde{m}_{\alpha_1,t}/\partial\hat{v}_{1,s}, \partial\tilde{m}_{\alpha_2,t}/\partial\hat{v}_{2,s}$, and $\partial\tilde{m}_{\beta_{kl},t}/\partial\hat{v}_{kl,s}$. The recursive equations can be satisfied iteratively to obtain estimates of the conditional means and variances $\tilde{m}_{\alpha_1,t}, \tilde{V}_{\alpha_1,t}, \tilde{m}_{\alpha_2,t}, \tilde{V}_{\alpha_2,t}, \tilde{m}_{\beta_{kl},t}, \tilde{V}_{\beta_{kl},t}$ of the Gaussian variational parameters. The update equations for the remaining static variational parameters is expressed in terms of these Gaussian parameters. Hence, following update of these parameters using the Kalman filtering technique described above, the updates for the rest of the parameters can be obtained using lemma 3.

D Updates for Special Distributions

The following tables give updates for some special cases, often encountered in real affinity estimation applications. For entity attributes, the updates can be obtained by plugging in the suitable inverse cummulant functions for the updates of the corresponding natural parameters $\boldsymbol{\theta}_{1k}, \boldsymbol{\theta}_{2l}$ of the family. Similarly, suitable GLM regression and the required expected values of the missing affinities is given in table 2.

Table 1: Important special case distributions for entity attributes

Distribution	$\psi(\theta)$	$\nabla\psi(\theta)$	$\nabla\psi^{-1}(t)$
Bernoulli	$\log(1+\exp(\theta))$	$(1+\exp(\theta))^{-1}$	$\log\left(\frac{t}{1-t}\right)$
Binomial	$N\log(1+\exp(\theta))$	$N(1+\exp(\theta))^{-1}$	$\log\left(\frac{t}{N-t}\right)$
Poisson	$\exp(\theta)-1$	$\exp(\theta)$	$\log t$
Gaussian	$\frac{\theta^2}{2}$	θ	t
Gamma	$-\log(-\theta)$	$\frac{1}{\theta}$	$\frac{1}{t}$

Table 2: Important special case distributions for affinities

Distribution	$\boldsymbol{\beta}_{kl}$ Update	$\mathbb{E}[y_{mn}]$
Gaussian	Weighted least squares	$\left(\sum_{k,l=1}^{K,L} r_{1mk}r_{2nl}\boldsymbol{\beta}_{kl}^{\dagger}\boldsymbol{x}_{mn}\right)$
Bernoulli	Newton Raphson's method	$\left(1+\exp\left(\sum_{k,l=1}^{K,L} r_{1mk}r_{2nl}\boldsymbol{\beta}_{kl}^{\dagger}\boldsymbol{x}_{mn}\right)\right)^{-1}$
Poisson	Newton Raphson's method	$\exp\left(\sum_{k,l=1}^{K,L} r_{1mk}r_{2nl}\boldsymbol{\beta}_{kl}^{\dagger}\boldsymbol{x}_{mn}\right)$
Binomial	Newton Raphson's method	$N'\left(1+\exp\left(\sum_{k,l=1}^{K,L} r_{1mk}r_{2nl}\boldsymbol{\beta}_{kl}^{\dagger}\boldsymbol{x}_{mn}\right)\right)^{-1}$

E Latent Variables based Bayesian Affinity Estimation

A taxonomy on some of the related work for Bayesian affinity estimation involving latent variables can be described as follows.

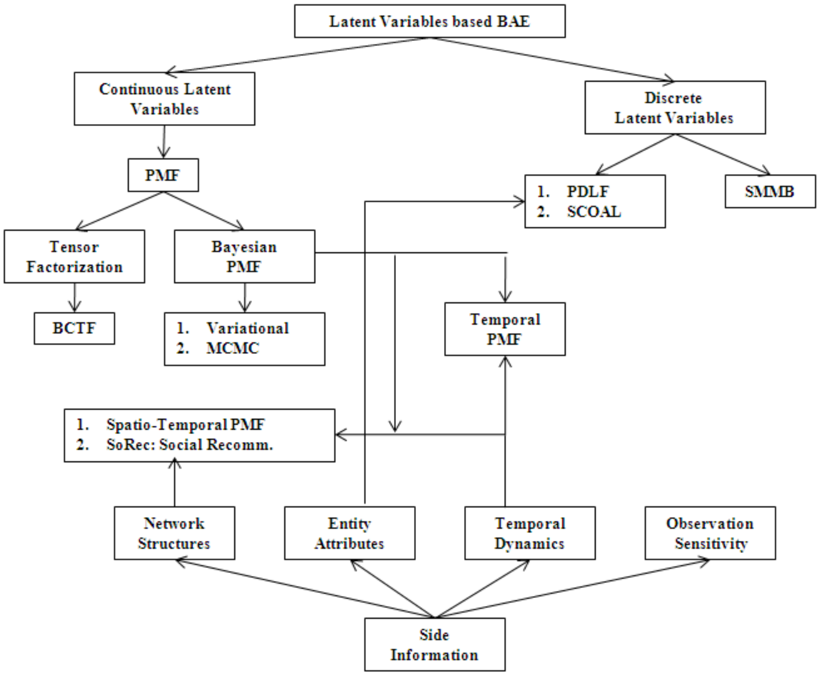

Figure 8: Latent Variables based Bayesian Affinity Estimation