Predictive Discrete Latent Factor Models for Large Scale Dyadic Data

Deepak Agarwal, Srujana Merugu
Yahoo! Research
Sunnyvale,CA,USA
{dagarwal,srujana}@yahoo-inc.com

ABSTRACT

We propose a novel statistical method to predict large scale dyadic response variables in the presence of covariate information. Our approach simultaneously incorporates the effect of covariates and estimates local structure that is induced by interactions among the dyads through a discrete latent factor model. The discovered latent factors provide a predictive model that is both accurate and interpretable. We illustrate our method by working in a framework of generalized linear models, which include commonly used regression techniques like linear regression, logistic regression and Poisson regression as special cases. We also provide scalable generalized EM-based algorithms for model fitting using both "hard" and "soft" cluster assignments. We demonstrate the generality and efficacy of our approach through large scale simulation studies and analysis of datasets obtained from certain real-world movie recommendation and internet advertising applications.

Categories and Subject Descriptors

H.1.1 [**Information Systems**]: Models and Principles

General Terms

Algorithms, Theory, Experimentation

Keywords

Generalized linear regression, Co-clustering, Latent factor modeling, Dyadic data

1. INTRODUCTION

Predictive modeling for *dyadic* data is an important data mining problem encountered in several domains such as social networks, recommendation systems, internet advertising, etc. Such problems involve measurements on *dyads*, which are pairs of elements from two different sets. Often, a response variable y_{ij} attached to dyads (i, j) measures interactions among elements in these two sets. Frequently, accompanying these response measurements are vectors of covariates $\mathbf{x}_{ij}$ that provide additional information which may help in predicting the response. These covariates could be specific to individual elements in the sets or to pairs from the two sets. In most large scale applications, the data is sparse, high dimensional (i.e., large number of dyads), noisy, and heterogeneous; this makes statistical modeling a challenging task. We elucidate further with a real-world example.

Consider an online movie recommendation application such as NetFlix, which involves predicting preference ratings of users for movies. This preference rating can be viewed as a dyadic response variable y_{ij}; it depends both on the user i and the movie j and captures interactions that exist among users and movies. Since both user and movie sets are large, the number of possible dyads is astronomical. However, most users rate only a small subset of movies, hence measurements (actual ratings provided by a user) are available only for a small fraction of possible dyads. In addition to the known user-movie ratings, there also exists other predictive information such as demographic information about users, movie content and other indicators of user-movie interactions, e.g., is the user's favorite actor part of the movie cast? These predictive factors can be represented as a vector of covariates $\mathbf{x}_{ij}$ associated with user-movie dyad (i, j). Incorporating covariate information in the predictive model may improve performance in practice. It is also often the case that some latent unmeasured characteristics that are not captured by these covariates induce a local structure in our dyadic space (e.g., spatial correlations induced due to cultural similarities). The main contribution of this paper is to show that accounting for such local structures directly in the predictive model along with information in the covariates often leads to better predictions. In fact, the local structures in some cases may provide additional insights about the problem and may lead to models that are both accurate and interpretable.

The predictive problem discussed above is not specific to movie recommendation systems and arises in several other contexts.(e.g., click rate estimation for webpage-ad dyads in internet advertising, estimating probabilities of a call between telephone dyads in telecommunication networks, etc.) Prior work provide solutions using both supervised and unsupervised learning approaches. The supervised learning approach involves building a regression or a classification model to predict the dyadic response y_{ij} solely as a function of the available covariates $\mathbf{x}_{ij}$. It has been well-studied with considerable literature on selecting informative covariates and obtaining bounds on the generalization error [18]. However, in general, this approach disregards any local structure that might be induced on the dyadic space due to other latent unmeasured factors. In contrast, the unsupervised approach focuses exclusively on capturing local structures in the response measurements on dyads. The discovered latent structures (e.g., clusters, principal components) provide insights about the interactions in the dyadic space which

KDD'07, August 12–15, 2007, San Jose, California, USA.
Copyright 2007 ACM 978-1-59593-609-7/07/0008 ...$5.00.

are useful in the absence of informative covariates. In fact, these local structures provide a parsimonious model for succinctly capturing the interactions in the dyadic matrix. However, since this approach does not *adjust* for the effects of covariates, the resulting latent structure may contain redundant information.

In this paper, we propose a statistical method that combines the benefits of both supervised and unsupervised learning approaches; we simultaneously incorporate the effect of covariates as in supervised learning and also account for any local structure that may be present in the data as in unsupervised learning. To achieve this, we model the response as a function of both covariates (captures global structure) and a discrete number of latent factors (captures local structure). Referring elements in the two sets that form the dyads as rows and columns, our model assumes that the row and column elements are separately assigned to a finite number of row and column clusters (or factors). The cross-product of these row and column clusters partition the dyadic matrix into a small number of rectangular *block* clusters; these provide an estimate of our latent factors. The row-column decoupling strategy provides an efficient algorithm to estimate latent structures by iteratively performing separate row and column clusterings.

To provide further intuition about our method, we note that when the assignments are exclusive (i.e., "hard") as opposed to probabilistic (i.e., "soft"), each row and column is assigned to one and only one row and column cluster respectively. This partitions the dyadic matrix into a small number of rectangular blocks or co-clusters. In this case, the covariate information and local structures are incorporated simultaneously by assuming that the mean (or some function of the mean) of the response variable is a sum of some unknown function of the covariates and a block-specific constant; both of which get estimated from the data. We note that for models solely based on covariates, the additional block-specific constant that is extracted by our method is assumed to be part of the noise model; by *teasing out* this extra information parsimoniously through a piecewise constant function, we provide a model that may lead to better generalization in practice. Furthermore, the estimated blocks and the corresponding constants are often representative of some latent unmeasured factors that contributes to the interactions seen in our dyadic matrix. For instance, cultural preferences may cause users in a certain geographic region to provide higher ratings to certain class of movies. The clusters obtained from our method when subjected to further analysis and follow-ups with domain experts may discover such patterns. Thus, our model is both accurate in terms of predictions and interpretable in terms of the clusters obtained.

To illustrate our methodology, we confine ourselves to the framework of generalized linear models (GLMs), which provides a flexible class of predictive methods based on exponential families. This class includes Gaussian, Poisson and Bernoulli distributions as special cases. Further, for this special class of statistical models, we model the latent factors through an approach that is related to co-clustering using Bregman divergences. The key step in our methodology is to find a co-clustering that provides the *best predictive performance after adjusting for the covariates*; this is accomplished through an iterative model fitting process in the generalized EM framework.

1.1 Key Contributions

This paper provides a predictive modeling approach for dyadic data that simultaneously exploits information in the available covariates and the local structure present in the dyadic response matrix. In particular, the current work makes the following key contributions.

Exponential Family	**PDF**	**Natural parameter θ**	**Cumulant $\psi(\theta)$**
Gaussian	$\frac{1}{\sqrt{(2\pi\sigma^2)}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$	$\frac{\mu}{\sigma}2$	$\frac{\sigma^2}{2}\theta^2$
Poisson	$\frac{\lambda^x e^{-\lambda}}{x!}$	$\log\lambda$	e^θ
Bernoulli	$p^x(1-p)^{(1-x)}$	$\log\left(\frac{p}{1-p}\right)$	$\log(1+e^\theta)$

Table 2.1: Examples of exponential families and associated parameters and cumulant functions. The natural statistic $t(x) = x$ for all three cases and σ is assumed to be constant.

- We present a novel method to model dyadic response as a function of available predictor information and unmeasured latent factors through a *predictive discrete latent factor model* (PDLF hereafter).

- We provide a model-based solution in the framework of generalized linear models (GLMs), which constitute a broad and flexible family of predictive models based on exponential families. In fact, it includes the widely used least-squares regression and logistic regression techniques as special cases.

- We propose a scalable, generalized EM-based algorithms for “soft” and “hard” assignments, that are linear in the number of non-zeros in the dyadic matrix. The algorithms generalize several existing algorithms including GLM regression [16], co-clustering using Bregman divergences [2], cross-association learning [4], NPMLE [1], etc.

- We present an extensive empirical evaluation of our procedure through simulation experiments, analysis of a publicly available movie rating dataset, and illustrations on a real dataset from an internet advertising application. We show that the PDLF model provides better prediction results and additional insights about the data in the form of highly interpretable clusters or latent factors.

2. PRELIMINARIES

We begin with a brief review of (i) one parameter exponential families, generalized linear regression models, and (ii) co-clustering on dyadic data.

2.1 Exponential Families.

One-parameter exponential families provide a coherent framework to study commonly occurring prediction problems with univariate response. A random variable X with density $f(x;\theta)$ is said to belong to a one-parameter exponential family if

$$f(x;\theta) = \exp(\theta t(x) - \psi(\theta))p_0(x)\ . \qquad (2.1)$$

Here, the unknown parameter (also called the natural parameter) $\theta \in \Theta$; $p_0(x)$ is a probability measure that does not depend on θ; $\psi(\theta)$ is the cumulant generating function of X[1], $t(x)$ is some function of x (in most examples, $t(x) = x$). In fact, $E(t(X)) = \psi^{'}(\theta)$ and $Var(t(X)) = \psi^{''}(\theta)$. Table 2.1 shows three important examples of exponential distributions and the associated parameters and cumulant functions.

[1] To keep the exposition simple, dispersion parameter is assumed to be 1.

GLM	Response Type	Link Function $g(y)$	Exponential Family
Least-squares Regression	$y \in \mathbb{R}$	y	Gaussian
Poisson Regression	$y \in \mathbb{Z}_{++}$	$log(y)$	Poisson
Logistic Regression	$y \in \{0, 1\}$	$log\left(\frac{y}{1-y}\right)$	Bernoulli

Table 2.2: Examples of generalized linear models for different types of response variables.

2.2 Generalized Linear Models.

Generalized linear models (GLM) provides an abstract framework to study classification and regression problems that are commonly encountered in practice. Least squares regression for continuous response and logistic regression for binary response are special cases. A GLM is characterized by two components.

(i) The distribution of the response variable Y belongs to a member of the exponential family as defined in equation 2.1 with examples provided in Table 2.1.

(ii) The mean $\mu(\theta) = \psi^{'}(\theta)$ is some unknown function of the predictor vector $\mathbf{x}$, i.e., $\mu(\theta) = g^{-1}(\mathbf{x}; \boldsymbol{\beta})$ for some unknown vector $\boldsymbol{\beta}$. The most common choice is to assume g is a function of $\mathbf{x}^t\boldsymbol{\beta}$. The function g which ensures that $g(\mu)$ is a linear function of the predictors is often referred to as a link function and the choice of g that ensures $\theta = \mathbf{x}^t\boldsymbol{\beta}$ is called the canonical link function. For instance, in the case of a Bernoulli distribution, $g(\mu) = \log(\mu/(1-\mu))$. Table 2.2 provides examples of canonical link functions for common exponential family members. [16] provides an excellent introduction to GLMs. Unless otherwise mentioned, we will only consider canonical link functions in our subsequent discussions.

Thus, if the response Y follows a GLM, the conditional density $f(y; \boldsymbol{\beta}^t\mathbf{x})$ of y given $\mathbf{x}$ depends on the unknown parameter $\boldsymbol{\beta}$ only through the linear function $\boldsymbol{\beta}^t\mathbf{x}$. Although predictive methods based on GLMs are in general effective, they fail to account for unobserved interactions that are often present in dyadic data after adjusting for the covariates; our method provides a solution to this problem. Before proceeding further, we provide background material on matrix co-clustering, which is closely related to our method. In fact, our method captures unaccounted interactions by performing co-clustering in a latent space through a mixture model.

2.3 Matrix Co-clustering

Co-clustering, or simultaneous clustering of both rows and columns, has become a method of choice for analyzing large and sparse data matrices[15, 2] due to its scalability and has been shown to be effective for predicting missing values in dyadic data exploiting the interactions that are often present in the observed response values. In particular, the Bregman co-clustering framework proposed in [2], presents a formulation from a matrix approximation point of view, wherein the row and column clusterings are chosen so as to minimize the error between the original matrix $\mathbf{Y}$ and a reconstructed matrix $\hat{\mathbf{Y}}$ (called the minimum Bregman information matrix) that depends only on the co-clustering, and certain summary statistics of $\mathbf{Y}$, e.g., co-cluster means. This formulation allows the approximation error to be measured as the weighted sum of element-wise Bregman divergence between the matrices $\mathbf{Y}$ and $\hat{\mathbf{Y}}$. This co-clustering formulation also permits an alternate interpretation in terms of a structured mixture model as presented in [17]. We briefly describe this connection.

For dyad (i, j), let $\rho(i)$ and $\gamma(j)$ denote the row and column membership of the i^{th} row and j^{th} column respectively. We assume the cluster ids for rows and columns belong to the sets $\{I : I = 1, \cdots, k\}$ and $\{J : J = 1, \cdots, l\}$ respectively. Whenever appropriate, I and J would be used as shorthand to mean $\rho(i) = I$ and $\gamma(j) = J$ respectively. Now, consider a mixture model given by

$$p(y_{ij}) = \sum_{I,J} p(I, J)p(y_{ij}|I, J) = \sum_{I,J} \pi_{I,J} f_\psi(y_{ij}; \theta_{i,j,I,J}) \tag{2.2}$$

where π_{IJ} denotes the prior probabilities associated with the latent variable pair (I, J) and $\theta_{i,j,I,J}$ is the corresponding natural parameter that could have additive structural constraints, e.g., $\theta_{i,j,I,J} = \theta_i + \theta_j + \theta_{I,J}$ (accommodates row, column and co-cluster interactions) or $\theta_{i,j,I,J} = \theta_{I,J}$ (accommodates only co-cluster interactions). Using the bijection result between (regular) exponential families and a special class of Bregman divergences [3] and the projection theorem characterizing the optimality of minimum Bregman information matrix with respect to generalized additive models in the natural parameter space [17], it can be shown that maximizing the log-likelihood of $\mathbf{Y}$ with respect to the appropriate choice of the mixture model eqn. (2.2) is analogous to minimizing the reconstruction error in the Bregman co-clustering framework. The mixture model, in general, results in soft cluster assignments and is exactly equivalent to the "hard" Bregman co-clustering formulation when the dispersion of the mixture components is assumed to be zero.

We note that conditional on the latent variables $\rho(i), \gamma(j)$, the mixture model in eqn. (2.2) captures interactions through the block[2] means; the main issue is to find an optimal clustering to adequately explain the local structure in our data. Also, omitting covariates may provide clusters that contain redundant information and inferior predictive performance; hence, the need to simultaneously adjust both for covariates and find an optimal clustering.

3. PREDICTIVE DISCRETE LATENT FACTOR MODEL

In this section, we describe our predictive discrete latent factor (PDLF) model for dyadic response that simultaneously incorporates information in the covariates within the GLM framework and accounts for unmeasured interactions via co-clustering methods. We also present a generalized EM algorithm to estimate the model parameters which is guaranteed to monotonically increase the marginal likelihood until it attains a local maximum.

Let $\mathbf{Y} = [y_{ij}] \in \mathbb{R}^{m \times n}$ denote the response matrix and let $\mathbf{X} = [\mathbf{x}_{ij}] \in \mathbb{R}^{m \times n \times s}$ denote the tensor corresponding to s prespecified covariates with $\mathbf{x}_{ij} \in \mathbb{R}^s$. Further, let $\mathbf{W} = [w_{ij}] \in \mathbb{R}^{m \times n}$ denote non-negative weights associated with the observations in $\mathbf{Y}$.[3]

Given $k \times l$ blocks (I, J) with prior probabilities π_{IJ}, the marginal distribution of response given covariates is given as

$$p(y_{ij}|\mathbf{x}_{ij}) = \sum_{I,J} \pi_{IJ} f_\psi(y_{ij}; \boldsymbol{\beta}^t\mathbf{x}_{ij} + \delta_{I,J}), \ [i]_1^m [j]_1^n, \tag{3.3}$$

[2] Henceforth, we refer to each mixture component as a block to maintain the analogy with the hard assignment case.

[3] In our examples, this is set to 1 for a valid observation and 0 for missing ones.

where f_ψ is an exponential family distribution with cumulant $\psi(\cdot)$, $\boldsymbol{\beta} \in \mathbb{R}^s$ denotes the regression coefficients associated with the pre-specified covariates, π_{IJ} denotes the prior and $\delta_{I,J}$ denotes the interaction effects associated with the block (I, J). Writing $\theta_{ij,IJ} = \boldsymbol{\beta}^t \mathbf{x}_{ij} + \delta_{I,J}$ and comparing with eqn. (2.2), we see the difference between the usual co-clustering models and PDLF. The latter is a richer class which performs co-clustering on the residuals after adjusting for the effect of covariates. Furthermore, the estimation of covariate effects and co-cluster means on the residuals are carried out simultaneously; the usual practice of *detrending* the data first to remove covariate effects and clustering the residuals may provide suboptimal results since the effects are not orthogonal. We note than an alternate way of forming a mixture distribution that is often pursued in the statistics literature is through a semi-parametric hierarchical model wherein $g(\mu_{ij}) = \boldsymbol{\beta}^t \mathbf{x}_{ij} + \delta_{ij}$, and δ_{ij}s follow a clustering model, namely, a mixture of distributions. For instance, if $y_{ij}|\delta_{ij} \sim N(\boldsymbol{\beta}^t \mathbf{x}_{ij} + \delta_{ij}, \sigma^2)$ and $\delta_{ij} \sim \sum_{i=1}^{k} \pi_i N(\mu_i, \tau^i)$, the marginal distribution of y_{ij} is a mixture of Gaussians given by $\sum_{p=1}^{k} \pi_k N(\boldsymbol{\beta}^t \mathbf{x}_{ij} + \mu_p, \sigma^2 + \tau^p)$ which is structurally similar to eqn. (3.3). However, such an approach does not exploit the special structure of the dyadic data which is done by the block model in eqn. (3.3). In particular, the block model assumes that block membership of dyadic elements can be completely specified in terms of row and column memberships in the corresponding row and column clusters respectively. This is the key feature of our method which makes it scalable; we express a two-dimensional clustering problem in terms of two iterative one dimensional clusterings. In fact, the co-clustering method could viewed as a process that iteratively clusters rows and columns; clustering on columns has a smoothing effect which enhances row clustering and vice versa. More specifically, there exist latent variables $\rho(i)$ and $\gamma(j)$ attached to the i^{th} row and j^{th} column which take values in the cluster membership sets $\{I : I = 1, \cdots, k\}$ (row clusters) and $\{J : J = 1, \cdots, l\}$ (column clusters). Thus, each observation is assumed to have been generated from a mixture distribution with $k \times l$ components, each of which corresponds to a particular choice of (I, J). Further, the mean function of each component distribution includes a term that models the dependence of response on covariates. Thus, the dyads (i, j) are assigned to blocks (I, J) (fractional for soft clustering, degenerate for hard clustering) and within each block, the mean is some global function of the covariates, but adjusted by block-specific off-sets $\{\delta_{I,J}\}$. Hence, we capture the local structure using a piecewise constant function with the row and cluster assignments imposing a block structure and simplifying the computations.

3.1 Generalized EM Algorithm.

We present a generalized EM algorithm to fit the mixture model in eqn. (3.3) to the data. Throughout, $\theta_{ij,IJ} = \boldsymbol{\beta}^t \mathbf{x}_{ij} + \delta_{I,J}$. Assuming the observations are all generated from eqn. (3.3) with weights given by $\mathbf{W}$, the incomplete data log-likelihood is given by

$$\begin{aligned} L(\boldsymbol{\beta}, \Delta, \Pi) &= \sum_{i,j} w_{ij} \log(p(y_{ij})) \\ &= \sum_{i,j} w_{ij} \log(\sum_{I,J} \pi_{IJ} f_\psi(y_{ij}; \theta_{ij,IJ})) \quad (3.4) \end{aligned}$$

where $\boldsymbol{\beta}$, $\Delta = \{\{\delta_{IJ}\}_{I=1}^{k}\}_{J=1}^{l}$ and $\Pi = \{\{\pi_{IJ}\}_{I=1}^{k}\}_{J=1}^{l}$ denote the model parameters. As in the case of simple mixture models, this data log-likelihood is not a convex function of the parameters $(\boldsymbol{\beta}, \Delta, \Pi)$ and cannot be readily optimized.

To facilitate maximization of log-likelihood defined in eqn. (3.4), we consider a complete data likelihood obtained by augmenting $\{y_{ij}\}_{ij}$ with the latent variables $\{\rho(i)\}_i$ and $\{\gamma(j)\}_j$. Following the analysis in [19], we consider the free-energy function, which is defined as the sum of the expected complete log-likelihood and the entropy of the latent variables with respect to an arbitrary distribution $\tilde{p}(\{\rho(i)\}_i, \{\gamma(j)\}_j)$.

Since $\{y_{ij}\}_{ij}$ are conditionally independent given the cluster assignments, which are themselves independent for $\{\rho(i), \gamma(j)\}_{ij}$ for different values of (i, j), it suffices to assume that

$$\tilde{p}(\{\rho(i)\}_i, \{\gamma(j)\}_j) = \prod_{ij} \tilde{p}_{ij}(\rho(i), \gamma(j)) \ .$$

Then, the free-energy function is given as

$$\begin{aligned} F(\boldsymbol{\beta}, \Delta, \Pi, \tilde{p}) &= \sum_{ij} w_{ij} E_{\tilde{p}_{ij}}[\log p(y_{ij}, \rho(i), \gamma(j))] + \\ & \quad \sum_{ij} w_{ij} H(\tilde{p}_{ij}), \qquad (3.5) \end{aligned}$$

where $E_{\tilde{p}_{ij}}[\log p(y_{ij}, \rho(i), \gamma(j))] = \sum_{IJ} \tilde{p}(I, J) \log(\pi_{IJ} f_\psi(y_{ij}; \theta_{ij,IJ}))$ and $H(\tilde{p}_{ij}) = -\sum_{IJ} \tilde{p}_{ij}(I, J) \log(\tilde{p}_{ij}(I, J))$.

As proved in [19], EM procedure can also be viewed as a greedy maximization approach where one alternates between maximizing F w.r.t. $\boldsymbol{\beta}, \Delta, \Pi$ for a fixed $\tilde{p}$ (call it the M-step) and maximizing $\tilde{p}$ for a fixed $\boldsymbol{\beta}, \Delta, \Pi$ (call it the E-step). This formulation of the EM algorithm leads to alternative maximization procedures. For instance, in our case, optimizing $\tilde{p}$ in terms of either $\{\rho(i)\}_i$ or $\{\gamma(j)\}_j$ holding the other fixed and alternating with the M-step would still increase the marginal likelihood at every iteration. In fact, the value of $\tilde{p}$ which maximizes F for fixed $\boldsymbol{\beta}, \Delta, \Pi$ is $P(\{\rho(i), \gamma(j)\}_{ij} | \{y_{ij}\}_{ij}, \boldsymbol{\beta}, \Delta, \Pi) = \prod_{ij} P(\rho(i), \gamma(j) | y_{ij}, \boldsymbol{\beta}, \Delta, \Pi)$, where

$$P(\rho(i) = I, \gamma(j) = J | y_{ij}, \boldsymbol{\beta}, \Delta, \Pi) \propto \pi_{IJ} f_\psi(y_{ij}; \theta_{ij,IJ})^{w_{ij}} \ .$$

This forms the basis of the classical EM algorithm in the context of mixture models but is too slow in practice for our problem, especially when the number of $\{y_{ij}\}$ gets large. To expedite computations, we confine ourselves to the class of $\tilde{p}_{ij}$ that factorize as, $\tilde{p}_{ij}(\rho(i), \gamma(j)) = \tilde{p}_i(\rho(i)) \tilde{p}_j(\gamma(j))$ in our generalized EM procedure. This implicitly assumes that $\rho(i)$ and $\gamma(j)$ are independent a-posteriori, an approximation that approaches the true posterior as the joint posterior of $\rho(i), \gamma(j)$ approaches degeneracy. The complete steps of the algorithm are given in table 1 and can be executed in *any* order. Under mild conditions, it can be shown that each of these steps monotonically increase the free energy function, with at least one step resulting in a strict increase, till a local optimum is attained. In particular, steps 4 and 5 in Algorithm 1 provide an iterative clustering scheme whereby rows are clustered exploiting the column clustering already obtained and vice versa. This characteristic of being able to assign each observed dyadic measurement to a block through a sequence of row and column clusterings is the key feature that makes our algorithm scalable and converge fast.

The generalized EM approach in Algorithm 1 provides closed form updates for the prior block probabilities $\{\pi_{IJ}\}$ and also the row and column cluster assignments, each of which only requires a computation time of $O(Nkl)$ per iteration, where N denotes the number of observations in $\mathbf{Y}$ (i.e., elements such that $w_{ij} \neq 0$). The regression coefficients $\boldsymbol{\beta}$ and interaction effects Δ, in general, do not have closed form updates, but can be readily computed using convex optimization methods such as the Newton-Raphson's method. In fact, since the generalized EM algorithm does not require an exact optimization over each argument [10], it is sufficient

Algorithm 1 Generalized EM Algorithm for PDLF Model

```
Input: Response matrix $\mathbf{Y} = [y_{ij}] \in \mathbb{R}^{m \times n}$ with measure $\mathbf{W} = [w_{ij}] \in [0,1]^{m \times n}$, covariates $\mathbf{X} = [\mathbf{x}_{ij}] \in \mathbb{R}^{m \times n \times s}$, exponential family with cumulant $\psi$, num. of row clusters $k$ and num. of row clusters $l$.
Output: Regression coefficients $\boldsymbol{\beta}$, Implicit interaction effects $\Delta$, Mixture component priors $\Pi$, latent variable assignments $\tilde{p}$ that (locally) optimize the objective function in eqn. (3.5).
Method:
  Initialize with arbitrary latent variable assignments $\tilde{p}$
  repeat
    Generalized M-Step
    Step 1: Update Priors: $\forall\ [I]_1^k, [J]_1^l$,
        $\pi_{IJ} \leftarrow \sum_{ij} w_{ij} \tilde{p}_i(I) \tilde{p}_j(J)$
    Step 2: Update Interaction Effects: $\forall\ [I]_1^k, [J]_1^l$
        $\delta_{IJ} \leftarrow \underset{\delta}{\operatorname{argmax}} \sum_{ij} w_{ij} \sum_{IJ} \tilde{p}_i(I)\tilde{p}_j(J) \left(y_{ij}\delta_{IJ} - \psi(\boldsymbol{\beta}^t \mathbf{x}_{ij} + \delta_{IJ})\right)$
    Step 3: Update Regression Coefficients:
        $\boldsymbol{\beta} \leftarrow \underset{\boldsymbol{\beta}}{\operatorname{argmax}} \sum_{ij} w_{ij} \sum_{IJ} \tilde{p}_i(I)\tilde{p}_j(J) \left(y_{ij}\boldsymbol{\beta}^t \mathbf{x}_{ij} - \psi(\boldsymbol{\beta}^t \mathbf{x}_{ij} + \delta_{IJ})\right)$
    Generalized E-Step
    Step 4: Update Row Cluster Assignments: $\forall\ [i]_1^m, [I]_1^k$,
        $\tilde{p}_i(I) \leftarrow c_i \left( \prod_{j,J} \left(\pi_{IJ} f_\psi(y_{ij}; \boldsymbol{\beta}^t \mathbf{x}_{ij} + \delta_{IJ})\right)^{w_{ij}\tilde{p}_j(J)} \right)^{\frac{1}{w_i}}$
        where $c_i$ is a normalizing factor s.t. $\sum_I \tilde{p}_i(I) = 1$ and $w_i = \sum_j w_{ij}$.
    Step 5: Update Column Cluster Assignments $\forall\ [j]_1^n, [J]_1^l$,
        $\tilde{p}_j(J) \leftarrow c_j \left( \prod_{i,I} \left(\pi_{IJ} f_\psi(y_{ij}; \boldsymbol{\beta}^t \mathbf{x}_{ij} + \delta_{IJ})\right)^{w_{ij}\tilde{p}_i(I)} \right)^{\frac{1}{w_j}}$
        where $c_j$ is a normalizing factor s.t. $\sum_J \tilde{p}_j(J) = 1$ and $w_j = \sum_i w_{ij}$.
  until convergence
  return $(\boldsymbol{\beta}, \Delta, \Pi, \tilde{p})$
  Predictive distribution for $(i,j)$: $\sum_{IJ} \tilde{p}_i(I)\tilde{p}_j(J) f_\psi(.; \boldsymbol{\beta}^{\mathbf{t}} \mathbf{x_{ij}} + \boldsymbol{\delta_{IJ}})$
```

to perform a few iterations of the Newton-Raphson's method, each of which requires a computation time of $O(N(kl + s^2))$. Thus, assuming a constant number of iterations, the overall algorithm only requires a computation time that is linear in the number of observations. For special cases such as Gaussian and Poisson distributions, it turns out that the interaction effects Δ can be computed in closed form as in Table 3.3. This is possible due to the functional form of the cumulant which is given by $\psi(x) \propto x^2$ for Gaussian and $\psi(x) \propto \exp(x)$ for the Poisson. For the Gaussian, the regression coefficients $\boldsymbol{\beta}$ can also be computed in closed form using a weighted least squares regression on the residuals $y_{ij} - \delta_{IJ}$.

4. HARD ASSIGNMENT PDLF MODEL

In this section, we analyze a special case of our latent factor model where each row (column) is exclusively assigned to a single latent factor, i.e., a row (column) cluster, and describe a highly scalable algorithm for this setting.

For the special case corresponding to hard assignments, the latent factor model in eqn. (3.3) can be expressed as

$$p(y_{ij}|\mathbf{x}_{ij}, \rho, \gamma) = f_\psi(y_{ij}; \boldsymbol{\beta}^t \mathbf{x}_{ij} + \delta_{\rho(i),\gamma(i)}),\ [i]_1^m [j]_1^n, \quad (4.6)$$

where the ij^{th} element is assigned exclusively to the block $(\rho(i), \gamma(j))$. For every block (I, J), let $\mathbf{X}^{latent^{I,J}}$ denote a binary-valued covariate that indicates if a dyad belongs to the IJ^{th} block, i.e.,

$$\begin{aligned} x_{ij}^{latent^{I,J}} &= 1,\ \text{when } I = \rho(i),\ J = \gamma(j) \\ &= 0,\ \text{otherwise.} \end{aligned}$$

We can now express the PDLF model in eqn. (4.6) as a generalized linear model over the initial set of covariates $\mathbf{X} \in \mathbb{R}^{m \times n \times s}$ and new set of latent covariates $\mathbf{X}^{latent} \in \mathbb{R}^{m \times n \times kl}$ associated with the $k \times l$ co-clusters, i.e.,

$$p(y_{ij}|\mathbf{x}_{ij}, \mathbf{x}_{ij}^{latent}) = f_\psi(y_{ij}; \boldsymbol{\beta}^t \mathbf{x}_{ij} + \Delta^t \mathbf{x}_{ij}^{latent}),\ [i]_1^m [j]_1^n\ , \quad (4.7)$$

with Δ being the coefficients of the covariates $\mathbf{X}^{latent}$. However, unlike in a simple generalized linear model, the covariates $\mathbf{X}^{latent}$ are not known beforehand. Hence, the learning procedure in this case, involves two steps:

(a) Discovering the "most informative" set of latent covariates of a specific form (binary-valued indicators of disjoint blocks of the response matrix), i.e., the best co-clustering (ρ, γ).

(b) Fitting a GLM over the combination of covariates in $\mathbf{X}$ and $\mathbf{X}^{latent}$.[4]

The above two steps, in fact, correspond to the generalized EM steps in Algorithm 1. To see the connection, consider the free energy function in eqn. (3.5). Since each row (column) is exclusively assigned to a single row (column) cluster, the conditional entropy term vanishes and there is also no dependency of the assignments on the priors of the mixture components. Hence, the free energy function (up to an additive constant) for the hard assignment case is given by

$$\begin{aligned} F^{hard}(\boldsymbol{\beta}, \Delta, \rho, \gamma) &= \sum_{ij} w_{ij} \log f_\psi(y_{ij}; \boldsymbol{\beta}^t \mathbf{x}_{ij} + \delta_{\rho(i),\gamma(j)}) \\ &= \sum_{ij} w_{ij} \log f_\psi(y_{ij}; \boldsymbol{\beta}^t \mathbf{x}_{ij} + \mathbf{x}_{ij}^{latent^t} \Delta) \\ &= F^{hard}(\boldsymbol{\beta}, \Delta, \mathbf{x}_{ij}^{latent})\ . \end{aligned} \quad (4.8)$$

As in the case of the general PDLF model in eqn. (4.6), the above objective function can be optimized by a repeatedly maximizing over the parameters $\boldsymbol{\beta}, \Delta$ and the cluster assignments (ρ, γ) (i.e., latent covariates $\mathbf{X}^{latent}$) until a local maximum of the likelihood function is attained. Algorithm 2 shows the detailed updates for this case.

Note that for any exponential family distribution f_ψ, the update steps for the regression coefficients $\boldsymbol{\beta}$ and interaction effects Δ in Algorithm 2 can be combined into a single GLM regression. Since each row (column) is assigned to single row (column) cluster, the cluster assignments can also be performed quite efficiently requiring a computation time of only $O(N(k + l))$ per iteration.

[4] Note that we need to ensure that the covariates in $[\mathbf{X}, \mathbf{X}^{latent}]$ are linearly independent, possibly by excluding some of the co-cluster covariates, in order that the model is not over-parameterized.

Exponential Family	$\boldsymbol{\beta}$ Update	Δ Update
Gaussian	Single least-squares regression	$\delta_{IJ} \leftarrow \frac{1}{\pi_{IJ}} \sum_{i,j} w_{ij}\tilde{p}_i(I)\tilde{p}_j(J)(y_{ij} - \boldsymbol{\beta}^t \mathbf{x}_{ij}),\ [I]_1^k, [J]_1^l$
Poisson	Newton-Raphson's method	$\delta_{IJ} \leftarrow \log\left(\frac{\sum_{i,j} w_{ij}\tilde{p}_i(I)\tilde{p}_j(J)y_{ij}}{\sum_{i,j} w_{ij}\tilde{p}_i(I)\tilde{p}_j(J)\boldsymbol{\beta}^t \mathbf{x}_{ij}}\right),\ [I]_1^k, [J]_1^l$
Bernoulli	Newton-Raphson's method	Newton-Raphson's method

Table 3.3: Update steps for the regression coefficients and interaction effects for important special cases.

```
Algorithm 2 Hard PDLF Algorithm
Input: Response matrix $\mathbf{Y} = [y_{ij}] \in \mathbb{R}^{m \times n}$ with measure $\mathbf{W} = [p_{ij}] \in [0,1]^{m \times n}$, covariates $\mathbf{X} = [\mathbf{x}_{ij}] \in \mathbb{R}^{m \times n \times s}$, exponential family with cumulant $\psi$, num. of row clusters $k$ and num. of row clusters $l$.
Output: Regression coefficients $\boldsymbol{\beta}$, implicit interaction effects $\Delta$, hard latent variable assignments $(\rho, \gamma)$ that (locally) optimize the objective function in eqn. (4.8).
Method:
  Initialize with arbitrary latent variable assignments $(\rho, \gamma)$
  repeat
    Generalized M-Step
    Step 1: Update Interaction Effects: $\forall\ [I]_1^k, [J]_1^l$,
      $\delta_{IJ} \leftarrow \operatorname{argmax}_{\delta} \sum_{i \in I, j \in J} w_{ij}\left(y_{ij}\delta - \psi(\boldsymbol{\beta}^t \mathbf{x}_{ij} + \delta)\right)$
    Step 2: Update Regression Coefficients:
      $\boldsymbol{\beta} \leftarrow \operatorname{argmax}_{\boldsymbol{\beta}} \sum_{ij} w_{ij}\left(y_{ij}\boldsymbol{\beta}^t \mathbf{x}_{ij} - \psi(\boldsymbol{\beta}^t \mathbf{x}_{ij} + \delta_{\rho(i)\gamma(j)})\right)$
    Generalized E-Step
    Step 3: Update Row Cluster Assignments: $\forall\ [i]_1^m$,
      $\rho(i) \leftarrow \operatorname{argmax}_{I}\left(\sum_{j} w_{ij}(y_{ij}\delta_{I\gamma(j)} - \psi(\boldsymbol{\beta}^t \mathbf{x}_{ij} + \delta_{I\gamma(j)}))\right)$
    Step 4: Update Column Cluster Assignments: $\forall\ [j]_1^n$,
      $\gamma(j) \leftarrow \operatorname{argmax}_{J}\left(\sum_{i} w_{ij}(y_{ij}\delta_{\rho(i)J} - \psi(\boldsymbol{\beta}^t \mathbf{x}_{ij} + \delta_{\rho(i)J}))\right)$
  until convergence
  return $(\boldsymbol{\beta}, \Delta, \rho, \gamma)$
  Predictive Distribution for dyad $(i, j)$: $f_\psi(.; \boldsymbol{\beta}^t \mathbf{x}_{\mathbf{ij}} + \delta_{\rho(\mathbf{i})\gamma(\mathbf{j})})$
```

4.1 Special Cases: GLM and Block Co-clustering

Since the PDLF model combines ideas from GLMs and co-clustering, one would naturally expect these two methods to be special cases of the generalized EM algorithm for PDLF.

GLM. When $k = l = 1$, the entire dyadic space forms a single co-cluster so that there do not exist any latent covariates. Hence, the model in eqn. (4.7) reduces to a simple GLM.

Co-clustering. In the absence of pre-specified covariates, the free energy function (up to an additive constant) in eqn. (4.8) reduces to

$$F^{hard}(\Delta, \rho, \gamma) = \sum_{ij} w_{ij} \log f_\psi(y_{ij}; \delta_{\rho(i),\gamma(j)})\ . \qquad (4.9)$$

Using the bijection between regular exponential families and Bregman divergences [3], we can further rewrite it as

$$F^{hard}(\Delta, \rho, \gamma) = -\sum_{ij} w_{ij} d_\phi(y_{ij}, \hat{y}_{\rho(i),\gamma(j)})\ , \qquad (4.10)$$

where d_ϕ is the Bregman divergence corresponding to the Legendre conjugate of ψ and $\hat{y}_{\rho(i),\gamma(j)} = \psi'(\delta_{\rho(i),\gamma(j)})$. The likelihood maximization problem can now be cast as minimizing the matrix approximation error with respect to the original response $\mathbf{Y}$ using a simple reconstruction based on block co-clustering (i.e., basis $\mathcal{C}_2$ in [2]).

5. EMPIRICAL EVALUATION

In this section, we provide empirical evidence to highlight the flexibility and efficacy of our PDLF approach. First, we describe controlled experiments on simulated data to analyze the predictive performance of our algorithms relative to other existing approaches. Then, we present results on real-world datasets for movie-recommendations (MovieLens)[9] and ad click-analysis (Yahoo! internal dataset) to demonstrate the benefits of our approach for a variety of learning tasks such as relevance classification, imputation of continuous missing values and feature discovery.

5.1 Simulation Studies on Gaussian Models

We first study the performance of our predictive modeling algorithms (Algorithms 1 and 2) on synthetic data generated from PDLF, and some simpler special cases of PDLF described in table 5.4.

Data Simulation. To choose realistic parameters for the generative models, we analyzed a subsample of the `MovieLens` dataset consisting of 168 users, 197 movies and 2872 ratings (response variable) as well as attributes based on user demographics (e.g., age/gender/occupation) and movie genres (e.g., science-fiction). From this dataset, we obtained four important covariates and computed the corresponding linear regression coefficients (i.e., $\boldsymbol{\beta}$) using a Gaussian linear model for the ratings. We also independently co-clustered the response matrix (assuming $k = l = 5$) without using the covariate information to obtain co-clusters, reasonable values for the co-clusters priors π, the row/column effects (say $\boldsymbol{\mu} \in \mathbb{R}^m$ and $\boldsymbol{\nu} \in \mathbb{R}^n$), and the co-cluster interaction effects (i.e., Δ). We consider five generative models based on various combinations of these parameters as shown in Table 5.4. In each case, we simulated 200 datasets from the model.[5]

5.1.1 Model Recovery using Soft and Hard Assignments.

For our first experiment, we used the 200 datasets generated from the PDLF model, i.e., the mixture of generalized linear models M_1. Our goal here is two-fold: a) To provide a sanity check on the PDLF model by fitting it to data where it should work and b)To compare

[5] The data and the models can be downloaded from `http://www.lans.ece.utexas.edušrujana`

Model	Parameter Constraints	Appropriate Algorithm
M_1	none	Soft PDLF Algorithm
M_2	$\boldsymbol{\mu} = \mathbf{0}, \boldsymbol{\nu} = \mathbf{0}, \Delta = \mathbf{0}$	Linear Regression
M_3	$\Delta = \mathbf{0}$	Linear Regression with row/col effects
M_4	$\boldsymbol{\beta} = \mathbf{0}, \boldsymbol{\mu} = \mathbf{0}, \boldsymbol{\nu} = \mathbf{0}$	Co-clustering
M_5	$\boldsymbol{\beta} = \mathbf{0},$	Co-clustering with row/col effects

Table 5.4: Generative models used for simulation studies

the effectiveness of the generalized EM (or "soft") algorithm (Algorithm 1) and the one that uses hard assignments (Algorithm 2) in estimating the true model parameters.

To each simulated data, we applied the PDLF algorithms corresponding to Gaussian distributions with $k = l = 5$. To avoid local optima, for each dataset, we repeated the algorithm with five different initializations and picked the best overall solution (we did not initialize with the true cluster assignments or true parameter values that were used in the simulations.) Table 5.5 show the true values of the covariate coefficients $\boldsymbol{\beta}$ and the 95% confidence intervals for the soft and hard PDLF algorithms. From the results, we observe that the true $\boldsymbol{\beta}$ values always lie in the 95% confidence interval for both the algorithms providing a sanity check on our code, model formulation and algorithms. In comparing the soft and hard PDLF algorithm, while the β values are similar (hard PDLF tends to have slightly higher variation in estimating β), the dispersion parameter or variance of the Gaussian distribution is underestimated by hard PDLF providing evidence of overfitting. The 95% confidence intervals for σ^2 obtained from the soft PDLF algorithm includes the truth. To avoid the overfitting problem with hard PDLF, we implemented a hybrid PDLF whereby we start out with a soft PDLF but switch to the hard one after a few iterations. say that this ameliorates the situation to some extent; recommended strategy if possible to implement.

Algo	β_0	β_1	β_2	β_3	β_4	σ^2
True	3.78	0.51	-0.28	0.14	0.24	1.16
Soft	(3.69,3.84)	(-0.31,0.71)	(-0.52,-0.19)	(-0.05,0.17)	(-0.64,1.04)	(1.14,1.27)
Hard	(3.66,3.84)	(-0.63,0.62)	(-0.58,-0.16)	(-0.09,0.18)	(-0.68,1.05)	(0.90,.99)

Table 5.5: 95% quantiles of the $\boldsymbol{\beta}$ values estimated using the "soft" and "hard" PDLF algorithms.

5.1.2 Robustness of PDLF Model.

Next, we consider the various special cases of the PDLF model in eqn. (4.6) that arise from disregarding the contributions of the covariates, row/col effects or the interaction effects as listed in Table 5.4. For each of these models, there exists a simpler learning approach that captures the associated structural assumptions. In this experiment, we study the predictive performance of our PDLF algorithm when data is generated from a simpler model. This provides an assessment of robustness and overfitting properties of the PDLF model. Table 5.6 shows the prediction error [6] (mean square error with five-fold cross validation) using different algorithms on data generated from models $M_1 - M_5$. From the table, we observe that for each model, the test error using the PDLF algorithm is comparable to that of the special case algorithm appropriate for the model. This provides evidence on the robustness of the PDLF model. In fact, it shows that the presence of a few irrelevant features does not hurt the performance of PDLF and makes it a general tool to analyze dyadic response data.

Algorithm	**Mean Sq. Error**
Soft PDLF	0.7175 ± 0.0030
Linear Regression	0.7221 ± 0.0031
Linear Regression with row/col effects	0.7332 ± 0.0032
Co-clustering	0.7252 ± 0.0031
Co-clustering with row/col effects	0.7316 ± 0.0032

Table 5.7: Prediction error (mean square error with 5-fold cross-validation) using different algorithms with partial covariate information. $k = l = 5$ where applicable.

5.2 Case Study 1: Relevance Classification using Logistic Model

In this study, we explore the benefits of our approach for relevance classification, which involves predicting a binary response (relevant or not) given a pair of objects that can be interpreted as the rows and columns of the response matrix. There are two objectives in conducting this experiment: a)We show an application of PDLF for binary response and b) We show that combining covariate information and modeling local structure leads to better predictive performance relative to methods that do not account for both these information simultaneously.

For our experiments, we used a subset of the MovieLens dataset consisting of 459 users, 1410 movies and 20000 ratings (range 1-5) as well 23 attributes based on user demographics/movie genres and their interactions. We binarized the response variable by choosing ratings > 3 as relevant and ratings ≤ 3 as not relevant. To predict this binary-valued response, we consider a PDLF model based on Bernoulli (or logistic) distributions. For scalability, we restrict ourselves to the hard PDLF algorithm (Algorithm 2) with a fairly small number of row/column clusters $k = l = 5$. To evaluate our approach, we compare it against two methods that have been previously used to analyze this data: a) *Logistic regression* which is a supervised learning method that only incorporates covariate effects and b) *cross-association learning* [4] which is an unsupervised approach to learn a dyadic matrix consisting of binary response variable for prediction purposes. Table 5.8 shows the misclassification error and Figure 5.1 shows the precision-recall curves obtained using the different methods. We find better performance with PDLF, proving the benefit of simultaneously incorporating both covariate and cluster information for building effective predictive models for dyadic data.

Baseline	Logistic Regression	Cross Associations	PDLF
0.44 ± 0.0004	0.41 ± 0.0005	0.41 ± 0.007	0.37 ± 0.005

Table 5.8: Misclassification error (5-fold cross-validation) on MovieLens data. We choose k=l=5 for the both PDLF and cross-association learning.

5.3 Case Study 2: Imputation of Missing Values using Gaussian Model

This experiment focuses on the case where the dyadic response is continuous and the learning task can be viewed as predicting missing values in a matrix. We used the same MovieLens dataset as in

[6] Note that it is not fair to compare the log-likelihood or training error since the different algorithms involve varying number of parameters.

Model	Soft PDLF	Linear Regression	Linear Regression with row/col effects	Co-clustering	Co-clustering with row/col effects
M_1	**1.1436 ± 0.0047**	1.1496 ± 0.0046	1.1488 ± 0.0050	1.1566 ± 0.0049	1.1520 ± 0.0043
M_2	0.7172 ± 0.0030	**0.7193 ± 0.0030**	0.7178 ± 0.0030	0.7286 ± 0.0030	0.7290 ± 0.0032
M_3	0.7178 ± 0.0034	0.7199 ± 0.0029	**0.7191 ± 0.0029**	0.7312 ± 0.0029	0.7337 ± 0.0032
M_4	1.1357 ± 0.0050	1.1485 ± 0.0045	1.1408 ± 0.0048	**1.1327 ± 0.0048**	1.1426 ± 0.0049
M_5	1.1456 ± 0.0044	1.1497 ± 0.0047	1.1471 ± 0.0049	1.1458 ± 0.0046	**1.1448 ± 0.0048**

Table 5.6: Prediction error (mean square error with 5-fold cross validation) using different algorithms on data generated from models $M_1 - M_5$. $k = l = 5$ where applicable

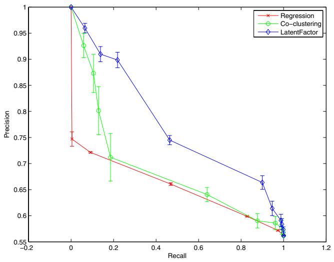


Figure 5.1: Precision-recall curves on MovieLens data. We choose k=l=5 for the both PDLF and cross-associations learning.

first case study. Since most of the existing techniques for addressing this task such as singular value decomposition(SVD) [8], nonnegative matrix factorization(NNMF) [13] and correlation-based methods [23] implicitly assume a Gaussian generative model, we transformed the response, i.e., the rating values using $y_{new} = \sqrt{(}6-y)$ to eliminate the skew and make the distribution more symmetric and close to Gaussian.

To predict this response, we use the hard PDLF algorithm (Algorithm 2) for Gaussian distributions with both row and column clusters set to 5; in addition we used covariates to account for the row and column effects. Table 5.9 shows the mean absolute error in the predictions (after inverse transformation) obtained using PDLF, k-rank SVD ($k = 5$), k-rank NNMF (squared loss, $k = 5$) bias adjusted co-clustering(COCLUST) (scheme $\mathcal{C}_5$,squared loss, $k = l = 5$) and simple linear regression (LINREG).

PDLF	LINREG	COCLUST	SVD	NNMF
0.80 ± 0.006	0.81 ± 0.006	0.83 ± 0.005	0.84 ± 0.004	0.83 ± 0.007

Table 5.9: Mean absolute error (5-fold cross-validation) on MovieLens data. We choose k=l=5 for the both PDLF and co-clustering and k=5 for SVD and NNMF.

As in the previous logistic regression example, we find that the PDLF model provides better predictive performance due of its flexibility to discover special clusters that have information not contained in the available covariates. For example, the PDLF model discovers a cluster containing not so well-known movies released in 1930's (shown in Table 5.10) while the co-clustering algorithm() without covariates only discovers groups that are predominantly characterized by the genre and rating levels, e.g. classic oscar-winning dramas. This demonstrates that other than providing accurate predictions, PDLF discovers clusters that are more informative.

Movies from the 30's (Sample movie cluster —PDLF)	Oscar winning dramas (Sample movie cluster —COCLUST)
Lost Horizon (1937)	Dead Man Walking
My Man Godfrey (1936)	Braveheart
Gay Divorcee, The (1934)	Dances with Wolves
Bride of Frankenstein (1935)	Godfather, The
Duck Soup (1933)	Silence of the Lambs, The

Table 5.10: Examples of movie clusters obtained using PDLF and direct co-clustering.

Cluster Id	Web-site clusters (rows)	Ip-domain clusters (columns)
1	shopping/search	Most non-clicking ips/US
2	popular shopping/search	aol/unknown/
3	aol/yahoo	educational/European
4	Most websites	Japanese
5	smaller portals	Korean

Table 5.11: Web-site and ip-domain clusters obtained using plain co-clustering

5.4 Case Study 3: Feature Discovery using Poisson Model

This experiment illustrates the utility of the proposed methodology for discovering hidden covariates. Specifically, we consider the task of predicting the number of times an ad served on a web-site is clicked from an ip (or ip-domain), which is useful for monitoring click volume and other related applications. For our experiment, we used a dataset consisting of 47903 ip-domains, 585 web-sites and 125208 ip-website dyads with click-counts and two covariates, ip-location and routing type. Since we deal with count data, we employ a PDLF model based on a Poisson distribution with $k = l = 5$. Similar to the earlier experiment, additional covariates that adjust for row(ip) and column(website) effects are also included. As in the previous two experiments, the predictive performance of the hard PDLF algorithm, measured in this case by I-divergence between observed and predicted (shown in Table 5.13) is better than a straightforward Poisson regression or the information-theoretic co-clustering [6] approach.

The clusters from the PDLF algorithm were rigorously analyzed. Figure 5.2 shows the co-clusters obtained before and after adjusting for the covariates and the row/column effects and the corresponding interaction effects. On examining the first co-clustering, we find that co-clusters (shown in Table 5.11) identify a number of highly predictive factors including the ip-domain location. In contrast, the PDLF approach reveals co-clusters (shown in Table 5.12 with a different set of interactions. In particular, the ip-domain clusters are no longer correlated with location and identify other interesting characteristics such as whether an ip-domain is a telecom company (column cluster 5) or a software/tech company (column cluster 3), which respectively happen to have positive interactions with internet portals (row cluster 4) and web media (row cluster 1).

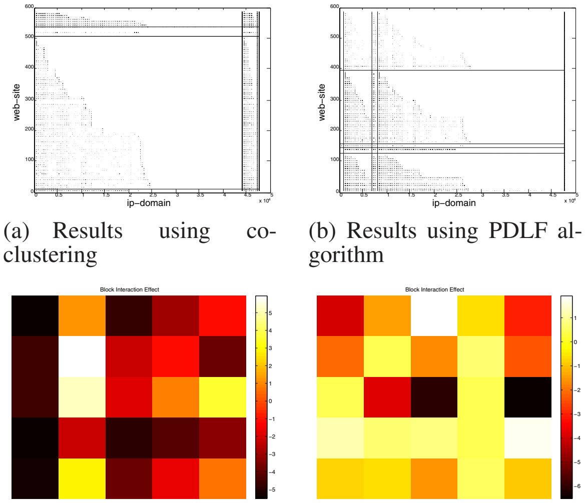
Cluster	Characteristic	Examples
Web-site cluster 1	Web Media	usatoday, newsgroups
Web-site cluster 4	Online Portals	msn, yahoo
Ip-domain cluster 3	Tech companies	agilent.com, intel.com
Ip-domain cluster 5	Telecom companies	sbcglobal.net, comcastbusiness.net

Table 5.12: Examples from web-site and ip-domain clusters obtained using PDLF.

From Section 4, we observe that the newly identified co-clusters can, in fact, be treated as new covariates allowing us to perform feature selection to obtain a model which generalize better. Table 5.13 (last column) shows that the predictive accuracy improves slightly after we eliminate some of the co-cluster based covariates.

PDLF	Linear Regression	COCLUST	PDLF with feature selection
54.09 ± 6.76	72.21 ± 0.94	77.72 ± 7.65	52.12 ± 2.44

Table 5.13: I-divergence loss (5-fold cross-validation) on click-count dataset. We choose k=l=5 for the both PDLF and co-clustering.



(a) Results using co-clustering

(b) Results using PDLF algorithm

(c) Interaction effects using co-clustering

(d) Interaction effects with PDLF model

Figure 5.2: Co-clusters obtained using direct information-theoretic co-clustering and the hard PDLF method and the corresponding interaction effects.

The proposed algorithm is quite efficient and can execute a single run of the algorithm (30 iterations) on this moderate sized dataset in about 40s in Matlab on a 1.86GHz Pentium M with 1GB RAM.

To briefly summarize the findings of this section. We provide sanity checks and a comparative analysis of soft and hard versions of PDLF through large scale simulations. We show both versions of the algorithm perform well with the hard version having a tendency to slightly overfit. We show that PDLF is robust in cases where a few covariates are not predictive and/or there is no local structure present in the data.

We conduct experiments on a publicly available MovieLens dataset using a logistic and Gaussian response model. We compare PDLF with existing supervised and unsupervised approaches that have been used to analyze this data and find superior performance. We also show that the clusters obtained from PDLF after adjusting for covariate effects are more informative. Finally, we conduct co-clustering analysis on a new real world dataset that is obtained from an application in internet advertising. The response variable in this case are click counts, hence we demonstrate PDLF on a Poisson model. This experiment is conducted on a much larger dataset and demonstrates the scalability of PDLF. Here again, simultaneous inclusion of both covariates and latent factors provides better performance relative to cases which does not include both. In fact, the cluster obtained for this experiment after adjusting for covariates are much more informative; the ones obtained without adjusting for covariates contain redundant information.

6. RELATED WORK

In this section, we briefly discuss how our PDLF model is related to existing literature in the machine learning and statistics communities. Our current work is primarily related to two active areas of research, namely (i) latent factor modeling of dyadic data, and (ii) hierarchical random effects modeling.

Latent Factor Modeling. In recent years, considerable research has been done on unsupervised learning methods in the context of dyadic data. Most methods of similar flavor such as singular value decomposition [8], non-negative matrix factorization [13],probabilistic latent semantic analysis [12], cross-association learning [4], Bregman co-clustering [2] are matrix approximation techniques, which impose different constraints on the latent structure depending on the choice of loss function. Among these approaches, co-clustering methods [15] based on iterative row and column clustering, have become popular due to their scalability. For a detailed survey on co-clustering methods, we refer the reader to [15]. We note that none of these methods make use of additional covariates for modeling the response as we do in our PDLF model.

Recently, Long et al. [14] proposed a relational summary network (RSN) model for clustering over k-partite graphs describing relations between k classes of entities. The RSN model considers not only pairwise interactions, but also allows for intrinsic attributes (covariates) associated with each entity. For the case $k = 2$, the data model associated with RSN (i.e., dyadic response and row/column predictors) is a special case of our data model. However, the RSN algorithm uses covariates only to influence the co-clustering, which is later used for predictive inference instead of directly leveraging the information contained in them. An important fact to note here is that in the RSN approach, the row and column clusters are chosen so as to be similar not only in terms of the associated dyadic responses, but also the associated covariates values, whereas in our approach, the co-clusters are forced to be maximally predictive of the response given the covariates.

Random Effects Modeling. The proposed PDLF model can also be interpreted as a statistical model that approximates local structure via a piecewise constant function in case of hard assignments. The mixture model formulation helps in smoothing out edge-effects in a hard cluster assignment model and provides better performance. An alternate strategy that has been widely used in the statistics literature provides a more continuous approximation through a hierarchical random effects model [22]. However, such models are mainly used for explanatory analysis and are not well suited for prediction tasks. Models similar to ours have been studied for small problems in one dimension [1]. More recently, [20] proposed a block model for binary dyadic data which models incidence matrices in social networks where both row and column elements are the same. However, their method does not incorporate covariates and was illustrated only on a small dataset. Another model of similar nature was proposed by [7] for spatial data. This method employs a one-dimensional discrete cluster model where the cluster assignment variables are modeled using a Potts allocation model.

Other Work. In the context of recommender systems, [21] considered combining information in the local structure of preference ratings as well as demographic and content-based covariates using an ensemble-based approach. This ensemble method, however, does not leverage the full potential of the underlying local structure and is not as interpretable as the PDLF model. Our current work is also related to recent work [5] on goal oriented or predictive clustering, which uses a bottleneck-like method, where the rows are clustered to retain maximal information about the dyadic response. Unlike our method, this approach only involves single-sided clustering and does not take into account additional covariates that might be available.

7. CONCLUSION

To summarize, our current work provides a fairly general and scalable predictive modeling methodology for large, sparse, dyadic data that simultaneously combines information from the available covariates and discovers local structure by using a statistical model-based approach that combines ideas from supervised and unsupervised learning. We prove the efficacy of our approach through simulation, analysis on a publicly available dataset and a new dataset in the domain of internet advertising. We find better predictive performance relative to simpler models and other existing approaches; we also demonstrate the interpretability of our approach by discovering meaningful clusters in our example datasets.

The hard PDLF approach, although scalable and fairly accurate in practice, showed signs of overfitting in our simulation experiments. We are currently exploring a hybrid algorithm which start with a hard PDLF, but switches to a soft PDLF after a few iterations. As is in the case of other statistical approaches, model and feature selection are critical to the predictive performance and these issues need to be further explored. We showed that our discrete latent factor model provide good approximations to account for the missing factors. However, this may involve choosing large values of k and l. An alternate strategy would be to work with a continuous latent factor model where the interactions are modeled through a distance function. Such strategies have been pursued recently for social network data [11]; generalization to dyadic data with elements obtained from two different sets is challenging. Although the current work focuses on predictive discrete latent factors based on generalized linear models, in principle, the proposed methodology could apply to non-linear predictive models where the mean is modeled using non-parametric methods like generalized additive models, splines, etc., but requires further investigation.

8. REFERENCES

[1] M. Aitkin. A general maximum likelihood analysis of overdispersion in generalized linear models. *Journal of Statistics and Computing*, 6(3):1573–1375, September 1996.

[2] A. Banerjee, I. Dhillon, J. Ghosh, S. Merugu, and D. Modha. A generalized maximum entropy approach to Bregman co-clustering and matrix approximation. *JMLR*, 2007. to appear.

[3] A. Banerjee, S. Merugu, I. Dhillon, and J. Ghosh. Clustering with Bregman divergences. *JMLR*, 6:1705–1749, 2005.

[4] D. Chakrabarti, S. Papadimitriou, D. Modha, and C. Faloutsos. Fully automatic cross-associations. In *KDD*, 2004.

[5] D. Chickering, D. Heckerman, C. Meek, J. C. Platt, and B. Thiesson. Targeted internet advertising using predictive clustering and linear programming. http://research.microsoft.com/ meek/papers/goal-oriented.ps.

[6] I. Dhillon, S. Mallela, and D. Modha. Information-theoretic co-clustering. In *KDD*, 2003.

[7] C. Fernandez and P. J. Green. Modelling spatially correlated data via mixtures: a Bayesian approach. *Journal of Royal Statistics Society Series B*, (4):805–826, 2002.

[8] G. Golub and C. Loan. *Matrix Computations*. John Hopkins University Press, Baltimore, MD., 1989.

[9] Movielens data set. http://www.cs.umn.edu/Research/GroupLens/data/ml-data.tar.gz.

[10] A. Gunawardana and W. Byrne. Convergence theorems for generalized alternating minimization procedures. *JMLR*, 6:2049–2073, 2005.

[11] P. Hoff, A. Raftery, and M. Handcock. Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97:1090–1098, 2002.

[12] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd Annual ACM Conference on Research and Development in Information Retrieval*, pages 50–57, Berkeley, California, August 1999.

[13] D. L. Lee and S. Seung. Algorithms for non-negative matrix factorization. In *NIPS*, pages 556–562, 2001.

[14] B. Long, X. Wu, Z. Zhang, and P. S. Yu. Unsupervised learning on k-partite graphs. In *KDD*, 2006.

[15] S. C. Madeira and A. L. Oliveira. Biclustering algorithms for biological data analysis: A survey. *IEEE Trans. Computational Biology and Bioinformatics*, 1(1):24–45, 2004.

[16] P. McCullagh and J. A. Nelder. *Generalized Linear Models*. Chapman & Hall/CRC, 1989.

[17] S. Merugu. *Distributed Learning using Generative Models*. PhD thesis, Dept. of ECE, Univ. of Texas at Austin, 2006.

[18] T. M. Mitchell. *Machine Learning*. McGraw-Hill Intl, 1997.

[19] R. Neal and G. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. In *Learning in Graphical Models*, pages 355–368. MIT Press, 1998.

[20] K. Nowicki and T. A. B. Snijders. Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association*, 96(455):1077–1087, 2001.

[21] M. Pazzani. A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review*, (5-6):393–408, 1999.

[22] J. Rasbash and H. Goldstein. Efficient analysis of mixed hierarchical and cross-classified random structures using a multilevel model. *Journal of Educational Statistics*, (4):337–350, 1994.

[23] P. Resnick, N. Iacovou, M. Suchak, P. Bergstorm, and J. Riedl. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In *Proceedings of the ACM Conference on CSCW*, pages 175–186, 1994.