
Probabilistic Relational PCA

Wu-Jun Li **Dit-Yan Yeung**
Dept. of Comp. Sci. and Eng.
Hong Kong University of Science and Technology
Hong Kong, China
{liwujun, dyyeung}@cse.ust.hk

Zhihua Zhang
School of Comp. Sci. and Tech.
Zhejiang University
Zhejiang 310027, China
zhzhang@cs.zju.edu.cn

Abstract

Principal component analysis (PCA) has been successfully used for data analysis in many applications. Probabilistic PCA (PPCA) provides a probabilistic formulation of PCA, bringing about additional advantages such as the ability of dealing with missing data. One crucial assumption made by both PCA and PPCA is that the instances are independent and identically distributed (i.i.d.). However, it is not uncommon for some real-world data, such as web pages and research papers, to contain relation (link) structure in addition to textual content information. Different instances in such data are correlated (linked) with each other, which implies that the common i.i.d. assumption is unreasonable for such relational data. In this paper, by explicitly modeling the covariance between instances as derived from the relational information, we propose a novel probabilistic dimensionality reduction method, called *probabilistic relational PCA* (PRPCA), for relational data analysis. Although the i.i.d. assumption is no longer adopted in PRPCA, the learning algorithms for PRPCA can still be devised easily like those for PPCA which makes explicit use of the i.i.d. assumption. Extensive experiments on some real-world data sets show that PRPCA can effectively utilize the relational information to dramatically outperform PCA and achieve state-of-the-art performance.

1 Introduction

Using a low-dimensional embedding to summarize a high-dimensional data set has been widely used for exploring the structure in the data. The methods for discovering such low-dimensional embedding are often referred to as dimensionality reduction (DR) methods. Principal component analysis (PCA) [14] is one of the most popular DR methods with great success in many applications. As a more recent development, probabilistic PCA (PPCA) [22] provides a probabilistic formulation of PCA [14] based on a Gaussian latent variable model [1]. Compared with the original non-probabilistic derivation of PCA in [13], PPCA possesses a number of practical advantages. For example, PPCA can naturally deal with missing values in the data; the expectation-maximization (EM) algorithm [10] used to learn the parameters in PPCA may be more efficient for high-dimensional data; it is easy to generalize the single model in PPCA to the mixture model case; furthermore, PPCA as a probabilistic model can naturally exploit Bayesian methods [2].

Like many existing DR methods, both PCA and PPCA are based on some assumptions about the data. One assumption is that the data should be represented as feature vectors all of the same dimensionality. Data represented in this form are sometimes referred to as *flat data* [11]. Another assumption is the so-called i.i.d. assumption, which means that the instances are assumed to be independent and identically distributed (i.i.d.).

However, the data in many real-world applications, such as web pages and research papers, contain relations or links between (some) instances in the data in addition to the textual content information which is represented in the form of feature vectors. Data of this sort, referred to as *relational*

*data*¹ [11, 21], can be found in such diverse application areas as web mining [4, 18, 24, 25], bioinformatics [23], social network analysis [5], and so on. *On one hand*, the link structure among instances cannot be exploited easily when traditional DR methods such as PCA are applied to relational data. Very often, the useful relational information is simply discarded. For example, a citation/reference relation between two papers provides very strong evidence for them to belong to the same topic even though they may bear low similarity in their content due to the sparse nature of the bag-of-words representation, but the relational information is not exploited at all when applying PCA or PPCA. One possible use of the relational information in PCA or PPCA is to first convert the link structure into the format of flat data by extracting some additional features from the links. However, as argued in [11], this approach fails to capture some important structural information in the data. *On the other hand*, the i.i.d. assumption underlying PCA and PPCA is unreasonable for relational data. In relational data, the attributes of the connected (linked) instances are often *correlated* and the class label of one instance may have an influence on that of a linked instance. For example, in biology, interacting proteins are more likely to have the same biological functions than those without interaction. Therefore, PCA and PPCA, or more generally most existing DR methods based on the i.i.d. assumption, are not suitable for relational data analysis.

In this paper, a novel probabilistic DR method called *probabilistic relational PCA* (PRPCA) is proposed for relational data analysis. By explicitly modeling the covariance between instances as derived from the relational information, PRPCA seamlessly integrates relational information and textual content information into a unified probabilistic framework. Two learning algorithms, one based on a closed-form solution and the other based on an EM algorithm [10], are proposed to learn the parameters of PRPCA. Although the i.i.d. assumption is no longer adopted in PRPCA, the learning algorithms for PRPCA can still be devised easily like those for PPCA which makes explicit use of the i.i.d. assumption. Extensive experiments on some real-world data sets show that PRPCA can effectively utilize the relational information to dramatically outperform PCA and achieve state-of-the-art performance.

In section 2, we introduce the notation and some basic background for matrix variate normal distributions. Section 3 presents a reformulation of PPCA based on matrix variate normal distributions to facilitate the subsequent presentation. We then present in section 4 our PRPCA model as a novel extension of PPCA for relational data. Experimental results are reported in section 5 and finally section 6 concludes our paper.

2 Notation and Preliminaries

In this section, we first introduce the notation that will be used in the rest of the paper. We then briefly review some basics about matrix variate normal distributions. More details can be found in [12].

2.1 Notation

We use boldface uppercase letters, such as \mathbf{K} , to denote matrices, and boldface lowercase letters, such as \mathbf{z} , to denote vectors. The i th row and the j th column of a matrix \mathbf{K} are denoted by \mathbf{K}_{i*} and \mathbf{K}_{*j} , respectively. K_{ij} denotes the element at the i th row and j th column of \mathbf{K} . z_i denotes the i th element of \mathbf{z} . \mathbf{K}^T is the transpose of \mathbf{K} , and \mathbf{K}^{-1} is the inverse of \mathbf{K} . $\mathbf{K} \succeq 0$ means that \mathbf{K} is positive semi-definite (psd) and $\mathbf{K} \succ 0$ means that \mathbf{K} is positive definite (pd). $\text{tr}(\cdot)$ denotes the trace of a matrix and $\text{etr}(\cdot) \triangleq \exp(\text{tr}(\cdot))$. $|\cdot|$ denotes the determinant of a matrix. \mathbf{I}_n is the identity matrix of size $n \times n$. \mathbf{e} is a vector of 1s, the dimensionality of which depends on the context. We overload $\mathcal{N}(\cdot)$ for both multivariate normal distributions and matrix variate normal distributions [12]. $\langle \cdot \rangle$ denotes the expectation operation and $\text{cov}(\cdot)$ denotes the covariance operation.

Note that in relational data, there exist both *content* and *link* observations. As in [22], $\{\mathbf{t}_n\}_{n=1}^N$ denotes a set of observed d -dimensional data (content) vectors, the $d \times q$ matrix \mathbf{W} denotes the q principal axes (or called factor loadings), $\boldsymbol{\mu}$ denotes the data sample mean, and $\mathbf{x}_n = \mathbf{W}^T(\mathbf{t}_n - \boldsymbol{\mu})$

¹In this paper, we use document classification as a running example for relational data analysis. Hence, for convenience of illustration, the specific term ‘textual content information’ is used in the paper to refer to the feature vectors describing the instances. However, the algorithms devised in this paper can be applied to any relational data in which the instance feature vectors can represent any attribute information.

denotes the corresponding q principal components (or called latent variables) of \mathbf{t}_n . We further use the $d \times N$ matrix \mathbf{T} to denote the content matrix with $\mathbf{T}_{*n} = \mathbf{t}_n$, and the $q \times N$ matrix \mathbf{X} to denote the latent variables of \mathbf{T} with $\mathbf{X}_{*n} = \mathbf{W}^T(\mathbf{t}_n - \boldsymbol{\mu})$. For relational data, the $N \times N$ matrix \mathbf{A} denotes the adjacency (link) matrix of the N instances. In this paper, we assume that the links are undirected. For those data with directed links, we will convert the directed links into undirected links which can keep the original physical meaning of the links. This will be described in detail in Section 4.1.1, and an example will be given in Section 5. Hence, $A_{ij} = 1$ if there exists a relation between instances i and j , and otherwise $A_{ij} = 0$. Moreover, we always assume that there exist no self-links, i.e., $A_{ii} = 0$.

2.2 Matrix Variate Normal Distributions

We now review some basics about matrix variate distributions [12] on which our PRPCA model is based.

Definition 1 [12] *The Kronecker product of two matrices \mathbf{P} ($m \times n$) and \mathbf{Q} ($p \times q$), denoted by $\mathbf{P} \otimes \mathbf{Q}$, is the $mp \times nq$ matrix defined by $\mathbf{P} \otimes \mathbf{Q} = (P_{ij} \mathbf{Q})$.*

Definition 2 [12] *For a matrix \mathbf{K} ($m \times n$), $\text{vec}(\mathbf{K})$ is the $mn \times 1$ vector defined as $\text{vec}(\mathbf{K}) = (\mathbf{K}_{*1}^T, \mathbf{K}_{*2}^T, \dots, \mathbf{K}_{*n}^T)^T$.*

Definition 3 [12] *The random matrix \mathbf{K} ($m \times n$) is said to follow a matrix variate normal distribution with mean matrix \mathbf{E} ($m \times n$) and covariance matrix $\boldsymbol{\Sigma} \otimes \boldsymbol{\Psi}$, where $\boldsymbol{\Sigma}$ ($m \times m$) $\succ 0$ and $\boldsymbol{\Psi}$ ($n \times n$) $\succ 0$, if $\text{vec}(\mathbf{K}^T) \sim \mathcal{N}_{mn}(\text{vec}(\mathbf{E}^T), \boldsymbol{\Sigma} \otimes \boldsymbol{\Psi})$. This is denoted as $\mathbf{K} \sim \mathcal{N}_{m,n}(\mathbf{E}, \boldsymbol{\Sigma} \otimes \boldsymbol{\Psi})$.*

Theorem 1 [12] *If $\mathbf{K} \sim \mathcal{N}_{m,n}(\mathbf{E}, \boldsymbol{\Sigma} \otimes \boldsymbol{\Psi})$, then the p.d.f. of \mathbf{K} is given by*

$$p(\mathbf{K}) = \frac{\text{etr} \left\{ -\frac{1}{2} \boldsymbol{\Sigma}^{-1} (\mathbf{K} - \mathbf{E}) \boldsymbol{\Psi}^{-1} (\mathbf{K} - \mathbf{E})^T \right\}}{(2\pi)^{mn/2} |\boldsymbol{\Sigma}|^{n/2} |\boldsymbol{\Psi}|^{m/2}}.$$

As said in [12], a matrix variate normal distribution arises when sampling from a multivariate normal population. For example, let $\mathbf{K}_{*1}, \dots, \mathbf{K}_{*n}$ be a random sample of size n from $\mathcal{N}_m(\mathbf{m}, \boldsymbol{\Sigma})$. The observation random matrix \mathbf{K} will follow the following distribution: $\mathbf{K} \sim \mathcal{N}_{m,n}(\mathbf{m}\mathbf{e}^T, \boldsymbol{\Sigma} \otimes \mathbf{I}_n)$.

Theorem 2 [12] *Let $\mathbf{K} \sim \mathcal{N}_{m,n}(\mathbf{0}, \boldsymbol{\Sigma} \otimes \boldsymbol{\Psi})$ where $\boldsymbol{\Sigma} = (\sigma_{ij})$ and $\boldsymbol{\Psi} = (\psi_{ij})$, then $\langle K_{i_1 j_1} K_{i_2 j_2} \rangle = \sigma_{i_1 i_2} \psi_{j_1 j_2}$.*

Theorem 3 [12] *Let $\mathbf{K} \sim \mathcal{N}_{m,n}(\mathbf{E}, \boldsymbol{\Sigma} \otimes \boldsymbol{\Psi})$ and \mathbf{F} is of size $n \times n$, then $\langle \mathbf{K} \mathbf{F} \mathbf{K}^T \rangle = \text{tr}(\mathbf{F}^T \boldsymbol{\Psi}) \boldsymbol{\Sigma} + \mathbf{E} \mathbf{F} \mathbf{E}^T$.*

3 Probabilistic PCA

To set the stage for the next section which introduces our PRPCA model, we first briefly present the derivation for PPCA [22], which was originally based on (vector-based) multivariate normal distributions, from the perspective of matrix variate normal distributions.

If we use $\boldsymbol{\Upsilon}$ to denote the Gaussian noise process and assume that $\boldsymbol{\Upsilon}$ and the latent variable matrix \mathbf{X} follow these distributions:

$$\boldsymbol{\Upsilon} \sim \mathcal{N}_{d,N}(\mathbf{0}, \sigma^2 \mathbf{I}_d \otimes \mathbf{I}_N), \quad (1)$$

$$\mathbf{X} \sim \mathcal{N}_{q,N}(\mathbf{0}, \mathbf{I}_q \otimes \mathbf{I}_N), \quad (2)$$

we can express a generative model as follows:

$$\mathbf{T} = \mathbf{W} \mathbf{X} + \boldsymbol{\mu} \mathbf{e}^T + \boldsymbol{\Upsilon}.$$

Based on some properties of matrix variate normal distributions in [12], we get the following results:

$$\begin{aligned} \mathbf{T} \mid \mathbf{X} &\sim \mathcal{N}_{d,N}(\mathbf{W} \mathbf{X} + \boldsymbol{\mu} \mathbf{e}^T, \sigma^2 \mathbf{I}_d \otimes \mathbf{I}_N), \\ \mathbf{T} &\sim \mathcal{N}_{d,N}(\boldsymbol{\mu} \mathbf{e}^T, (\mathbf{W} \mathbf{W}^T + \sigma^2 \mathbf{I}_d) \otimes \mathbf{I}_N). \end{aligned} \quad (3)$$

Let $\mathbf{C} = \mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I}_d$. The corresponding log-likelihood of the observation matrix \mathbf{T} is then

$$\mathcal{L} = \ln p(\mathbf{T}) = -\frac{N}{2} \left[d \ln(2\pi) + \ln |\mathbf{C}| + \text{tr}(\mathbf{C}^{-1}\mathbf{S}) \right], \quad (4)$$

where

$$\mathbf{S} = \frac{(\mathbf{T} - \boldsymbol{\mu}\mathbf{e}^T)(\mathbf{T} - \boldsymbol{\mu}\mathbf{e}^T)^T}{N} = \frac{\sum_{n=1}^N (\mathbf{T}_{*n} - \boldsymbol{\mu})(\mathbf{T}_{*n} - \boldsymbol{\mu})^T}{N}. \quad (5)$$

We can see that \mathbf{S} is just the sample covariance matrix of the content observations. It is easy to see that this log-likelihood form is the same as that in [22]. Using matrix notations, the graphical model of PPCA based on matrix variate normal distributions is shown in Figure 1(a) below.

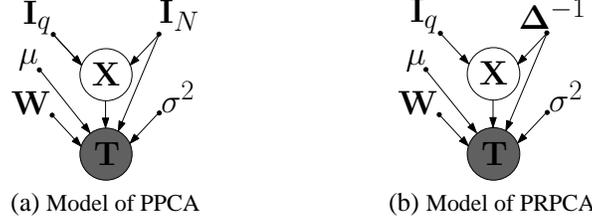


Figure 1: Graphical models of PPCA and PRPCA, in which \mathbf{T} is the observation matrix, \mathbf{X} is the latent variable matrix, $\boldsymbol{\mu}$, \mathbf{W} and σ^2 are the parameters to learn, and the other quantities are kept constant.

4 Probabilistic Relational PCA

PPCA assumes that all the observations are independent and identically distributed. Although this i.i.d. assumption can make the modeling process much simpler and has achieved great success in many traditional applications, this assumption is however very unreasonable for relational data [11]. In relational data, the attributes of connected (linked) instances are often *correlated*.

In this section, a probabilistic relational PCA model, called PRPCA, is proposed to integrate both the relational information and the content information seamlessly into a unified framework by eliminating the i.i.d. assumption. Based on our reformulation of PPCA using matrix variate notations as presented in the previous section, we can obtain PRPCA just by introducing some relatively simple (yet very effective) modifications. A promising property is that the computation needed for PRPCA is as simple as that for PPCA even though we have eliminated the restrictive i.i.d. assumption.

4.1 Model Formulation

Assume that the latent variable matrix \mathbf{X} has the following distribution:

$$\mathbf{X} \sim \mathcal{N}_{q,N}(\mathbf{0}, \mathbf{I}_q \otimes \boldsymbol{\Phi}). \quad (6)$$

According to Theorem 2, we can get $\text{cov}(\mathbf{X}_{i*}) = \boldsymbol{\Phi}$ ($i \in \{1, \dots, q\}$), which means that $\boldsymbol{\Phi}$ actually reflects the covariance between the instances. From (2), we can see that $\text{cov}(\mathbf{X}_{i*}) = \mathbf{I}_N$ for PPCA, which also coincides with the i.i.d. assumption of PPCA.

Hence, to eliminate the i.i.d. assumption for relational data, one direct way is to use a non-identity covariance matrix $\boldsymbol{\Phi}$ for the distribution of \mathbf{X} in (6). This $\boldsymbol{\Phi}$ should reflect the physical meaning (semantics) of the relations between entities, which will be discussed in detail later. Similarly, we can also change the \mathbf{I}_N in (1) to $\boldsymbol{\Phi}$ to eliminate the i.i.d. assumption for the noise process.

4.1.1 Relational Covariance Construction

Because the covariance matrix $\boldsymbol{\Phi}$ in PRPCA is constructed from the relational information in the data, we refer to it as *relational covariance* here.

The goal of PCA and PPCA is to find those principal axes onto which the retained variance under projection is maximal [14, 22]. For one specific \mathbf{X} , the retained variance is $\text{tr}[\mathbf{X}\mathbf{X}^T]$. If we rewrite (2) as

$$p(\mathbf{X}) = \frac{\exp\left\{\text{tr}\left[-\frac{1}{2}\mathbf{X}\mathbf{X}^T\right]\right\}}{(2\pi)^{qN/2}} = \frac{\exp\left\{-\frac{1}{2}\text{tr}[\mathbf{X}\mathbf{X}^T]\right\}}{(2\pi)^{qN/2}}, \quad (7)$$

we have the following observation:

Observation 1 For PPCA, the larger the retained variance of \mathbf{X} , i.e., the more \mathbf{X} approaches the destination point, the lower is the probability density at \mathbf{X} given by the prior.

Here, the *destination point* refers to the point where the goal of PPCA is achieved, i.e., the retained variance is maximal. Moreover, we use the retained variance as a *measure* to define the gap between two different points. The smaller is the gap between the retained variance of two points, the more they approach each other.

Because the design principle of PRPCA is similar to that of PPCA, our working hypothesis here is that Observation 1 can also guide us to design the relational covariance of PRPCA. Its effectiveness will be empirically verified in Section 5.

In PRPCA, we assume that the attributes of two linked instances are positively correlated.² Under this assumption, the *ideal goal* of PRPCA should be to make the latent representations of two instances as close as possible if there exists a relation (link) between them. Hence, the *measure* to define the gap between two points refers to the closeness of the linked instances, i.e., the summation of the Euclidean distances between the linked instances. Based on Observation 1, the more \mathbf{X} approaches the destination point, the lower should be the probability density at \mathbf{X} given by the prior. Hence, under the latent space representation \mathbf{X} , the closer the linked instances are, the lower should be the probability density at \mathbf{X} given by the prior. We will prove that if we set $\Phi = \Delta^{-1}$ where $\Delta \triangleq \gamma \mathbf{I}_N + (\mathbf{I}_N + \mathbf{A})^T (\mathbf{I}_N + \mathbf{A})$ with γ being typically a very small positive number to make $\Delta \succ 0$, we can get an appropriate prior for PRPCA. Note that $A_{ij} = 1$ if there exists a relation between instances i and j , and otherwise $A_{ij} = 0$. Because $\mathbf{A}^T = \mathbf{A}$, we can also express Δ as $\Delta = \gamma \mathbf{I}_N + (\mathbf{I}_N + \mathbf{A})(\mathbf{I}_N + \mathbf{A})$.

Let $\tilde{\mathbf{D}}$ denote a diagonal matrix whose diagonal elements $\tilde{D}_{ii} = \sum_j A_{ij}$. It is easy to prove that $(\mathbf{A}\mathbf{A})_{ii} = \tilde{D}_{ii}$. Let $\mathbf{B} = \mathbf{A}\mathbf{A} - \tilde{\mathbf{D}}$, which means that $B_{ij} = (\mathbf{A}\mathbf{A})_{ij}$ if $i \neq j$ and $B_{ii} = 0$. We can get $\Delta = (1+\gamma)\mathbf{I}_N + 2\mathbf{A} + \mathbf{A}\mathbf{A} = (1+\gamma)\mathbf{I}_N + \tilde{\mathbf{D}} + (2\mathbf{A} + \mathbf{B})$. Because $B_{ij} = \sum_{k=1}^N A_{ik}A_{kj}$ for $i \neq j$, we can see that B_{ij} is the number of paths, each with path length 2, from instance i to instance j in the original adjacency graph \mathbf{A} . Because the attributes of two linked instances are positively correlated, B_{ij} actually reflects the degree of correlation between instance i and instance j . Let us take the paper citation graph as an example to illustrate this. The existence of a citation relation between two papers often implies that they are about the same topic. If paper i cites paper k and paper k cites paper j , it is highly likely that paper i and paper j are about the same topic. If there exists another paper $a \neq k$ linking both paper i and paper j as well, the confidence that paper i and paper j are about the same topic will increase. Hence, the larger B_{ij} is, the stronger is the correlation between instance i and instance j . Because $B_{ij} = \sum_{k=1}^N A_{ik}A_{kj} = \mathbf{A}_{*i}^T \mathbf{A}_{*j}$, B_{ij} can also be seen as the similarity between the link vectors of instance i and instance j . Therefore, \mathbf{B} can be seen as a weight matrix (corresponding to a weight graph) derived from the original adjacency matrix \mathbf{A} , and \mathbf{B} is also consistent with the physical meaning underlying \mathbf{A} .

Letting $\mathbf{G} = 2\mathbf{A} + \mathbf{B}$,³ we can find that \mathbf{G} actually combines the original graph reflected by \mathbf{A} and the derived graph reflected by \mathbf{B} to get a new graph, and puts a weight $2A_{ij} + B_{ij}$ on the edge between instance i and instance j in the new graph. The new weight graph reflected by \mathbf{G} is also consistent with the physical meaning underlying \mathbf{A} . Letting $\mathbf{L} \triangleq \mathbf{D} - \mathbf{G}$, where \mathbf{D} is a diagonal matrix whose diagonal elements $D_{ii} = \sum_j G_{ij}$ and \mathbf{L} is called the Laplacian matrix [7] of \mathbf{G} , we can get $\Delta = (1+\gamma)\mathbf{I}_N + \tilde{\mathbf{D}} + \mathbf{D} - \mathbf{L}$. If we define another diagonal matrix $\hat{\mathbf{D}} \triangleq (1+\gamma)\mathbf{I}_N + \tilde{\mathbf{D}} + \mathbf{D}$, we can get $\Delta = \hat{\mathbf{D}} - \mathbf{L}$.

²Links with other physical meanings, such as the directed links in web graphs [26], can be transformed into links satisfying the assumption in PRPCA via some preprocessing strategies. One such strategy to preprocess the WebKB data set [9] will be given as an example in Section 5.

³This means that we put a 2:1 ratio between \mathbf{A} and \mathbf{B} . Other ratios can be obtained by setting $\Delta = \gamma \mathbf{I}_N + (\alpha \mathbf{I}_N + \mathbf{A})(\alpha \mathbf{I}_N + \mathbf{A}) = \gamma \mathbf{I}_N + \alpha^2 \mathbf{I}_N + 2\alpha \mathbf{A} + \mathbf{B}$. Preliminary results show that PRPCA is not sensitive to α as long as α is not too large, but we omit the detailed results here because they are out of the scope of this paper.

Then we have

$$\begin{aligned}
\text{tr}[\mathbf{X}\Delta\mathbf{X}^T] &= \text{tr}[\mathbf{X}\hat{\mathbf{D}}\mathbf{X}^T] - \text{tr}[\mathbf{X}\mathbf{L}\mathbf{X}^T] = \text{tr}[\mathbf{X}\hat{\mathbf{D}}\mathbf{X}^T] - \sum_{k=1}^q \mathbf{X}_{k*}\mathbf{L}\mathbf{X}_{k*}^T \\
&= \text{tr}[\mathbf{X}\hat{\mathbf{D}}\mathbf{X}^T] - \frac{1}{2} \sum_{k=1}^q \left[\sum_{i=1}^N \sum_{j=1}^N G_{ij} (X_{ki} - X_{kj})^2 \right] \\
&= \text{tr}[\mathbf{X}\hat{\mathbf{D}}\mathbf{X}^T] - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \left[G_{ij} \sum_{k=1}^q (X_{ki} - X_{kj})^2 \right] \\
&= \sum_{i=1}^N \hat{D}_{ii} \|\mathbf{X}_{*i}\|^2 - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N G_{ij} \|\mathbf{X}_{*i} - \mathbf{X}_{*j}\|^2. \tag{8}
\end{aligned}$$

Letting $\Phi = \Delta^{-1}$, we can get

$$p(\mathbf{X}) = \frac{\exp\left\{\text{tr}\left[-\frac{1}{2}\mathbf{X}\Delta\mathbf{X}^T\right]\right\}}{(2\pi)^{qN/2} |\Delta|^{-q/2}} = \frac{\exp\left\{-\frac{1}{2}\text{tr}[\mathbf{X}\Delta\mathbf{X}^T]\right\}}{(2\pi)^{qN/2} |\Delta|^{-q/2}}. \tag{9}$$

The first term $\sum_{i=1}^N \hat{D}_{ii} \|\mathbf{X}_{*i}\|^2$ in (8) can be treated as a measure of weighted variance of all the instances in the latent space. We can see that the larger \hat{D}_{ii} is, the more weight will be put on instance i , which is reasonable because \hat{D}_{ii} mainly reflects the degree of instance i in the graph. It is easy to see that, for those latent representations having a fixed value of weighted variance $\sum_{i=1}^N \hat{D}_{ii} \|\mathbf{X}_{*i}\|^2$, the closer the latent representations of two linked entities are, the larger is their contribution to $\text{tr}[\mathbf{X}\Delta\mathbf{X}^T]$, and subsequently the less is their contribution to $p(\mathbf{X})$. This means that under the latent space representation \mathbf{X} , the closer the linked instances are, the lower is the probability density at \mathbf{X} given by the prior. Hence, we can get an appropriate prior for \mathbf{X} by setting $\Phi = \Delta^{-1}$ in (6).

4.1.2 Model

With the constructed relational covariance Φ , the generative model of PRPCA is defined as follows:

$$\Upsilon \sim \mathcal{N}_{d,N}(\mathbf{0}, \sigma^2 \mathbf{I}_d \otimes \Phi) \tag{10}$$

$$\mathbf{X} \sim \mathcal{N}_{q,N}(\mathbf{0}, \mathbf{I}_q \otimes \Phi), \tag{11}$$

$$\mathbf{T} = \mathbf{W}\mathbf{X} + \boldsymbol{\mu}\mathbf{e}^T + \Upsilon,$$

where $\Phi = \Delta^{-1}$.

We can further obtain the following results:

$$\mathbf{T} \mid \mathbf{X} \sim \mathcal{N}_{d,N}(\mathbf{W}\mathbf{X} + \boldsymbol{\mu}\mathbf{e}^T, \sigma^2 \mathbf{I}_d \otimes \Phi), \tag{12}$$

$$\mathbf{T} \sim \mathcal{N}_{d,N}(\boldsymbol{\mu}\mathbf{e}^T, (\mathbf{W}\mathbf{W}^T + \sigma^2 \mathbf{I}_d) \otimes \Phi). \tag{13}$$

The graphical model of PRPCA is illustrated in Figure 1(b), from which we can see that the difference between PRPCA and PPCA lies solely in the difference between Φ and \mathbf{I}_N . Comparing (13) to (3), we can find that the observations of PPCA are sampled independently while those of PRPCA are sampled with correlation. In fact, PPCA may be seen as a degenerate case of PRPCA as detailed below in Remark 1:

Remark 1 When the i.i.d. assumption holds, i.e., all $A_{ij} = 0$, PRPCA degenerates to PPCA by setting $\gamma = 0$. Note that the only role that γ plays is to make $\Delta \succ 0$. Hence, in our implementation, we always set γ to a very small positive value, such as 10^{-6} . Actually, we may even set γ to 0, because Δ does not have to be pd. In Definition 3, $\Sigma \succ 0$ and $\Psi \succ 0$ can be relaxed to $\Sigma \succeq 0$ and $\Psi \succeq 0$. When $\Sigma \succeq 0$ and $\Psi \succeq 0$, we say \mathbf{K} follows a singular matrix variate normal distribution [12], and all the derivations for PRPCA are still correct. In our experiment, we find that the performance under $\gamma = 0$ is almost the same as that under $\gamma = 10^{-6}$. Further deliberation is out of the scope of this paper.

As in PPCA, we set $\mathbf{C} = \mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I}_d$. Then the log-likelihood of the observation matrix \mathbf{T} in PRPCA is

$$\mathcal{L}_1 = \ln p(\mathbf{T}) = -\frac{N}{2} \left[d \ln(2\pi) + \ln |\mathbf{C}| + \text{tr}(\mathbf{C}^{-1}\mathbf{H}) \right] + c, \quad (14)$$

where $c = -\frac{d}{2} \ln |\Phi|$ can be seen as a constant independent of the parameters $\boldsymbol{\mu}$, \mathbf{W} and σ^2 , and \mathbf{H} is defined as:

$$\mathbf{H} = \frac{(\mathbf{T} - \boldsymbol{\mu}\mathbf{e}^T)\boldsymbol{\Delta}(\mathbf{T} - \boldsymbol{\mu}\mathbf{e}^T)^T}{N}. \quad (15)$$

It is interesting to compare (14) with (4). We can find that to learn the parameters \mathbf{W} and σ^2 , the only difference between PRPCA and PPCA lies in the difference between \mathbf{H} and \mathbf{S} . Hence, all the learning techniques derived previously for PPCA are also potentially applicable to PRPCA simply by substituting \mathbf{S} with \mathbf{H} .

4.2 Learning

By setting the gradient of \mathcal{L}_1 with respect to $\boldsymbol{\mu}$ to 0, we can get the maximum-likelihood estimator (MLE) for $\boldsymbol{\mu}$ as follows:

$$\boldsymbol{\mu} = \frac{\mathbf{T}\boldsymbol{\Delta}\mathbf{e}}{\mathbf{e}^T\boldsymbol{\Delta}\mathbf{e}}. \quad (16)$$

As in PPCA [22], we devise two methods to learn \mathbf{W} and σ^2 in PRPCA, one based on a closed-form solution and the other based on EM.

4.2.1 Closed-Form Solution

Theorem 4 *The log-likelihood in (14) is maximized when*

$$\mathbf{W}_{ML} = \mathbf{U}_q(\boldsymbol{\Lambda}_q - \sigma_{ML}^2\mathbf{I}_q)^{1/2}\mathbf{R}, \quad (17)$$

$$\sigma_{ML}^2 = \frac{\sum_{i=q+1}^d \lambda_i}{d-q}, \quad (18)$$

where $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_d$ are the eigenvalues of \mathbf{H} , $\boldsymbol{\Lambda}_q$ is a $q \times q$ diagonal matrix containing the first q largest eigenvalues, \mathbf{U}_q is a $d \times q$ matrix in which the q column vectors are the principal eigenvectors of \mathbf{H} corresponding to $\boldsymbol{\Lambda}_q$, and \mathbf{R} is an arbitrary $q \times q$ orthogonal rotation matrix.

The proof of Theorem 4 makes use of techniques similar to those in Appendix A of [22] and is omitted here.

4.2.2 EM Algorithm

During the EM learning process, we treat $\{\mathbf{W}, \sigma^2\}$ as parameters, \mathbf{X} as missing data and $\{\mathbf{T}, \mathbf{X}\}$ as complete data. The EM algorithm operates by alternating between the E-step and M-step. Here we only briefly describe the update rules and their derivation is left to Appendix A.

In the E-step, the expectation of the complete-data log-likelihood with respect to the distribution of the missing data \mathbf{X} is computed. To compute the expectation of the complete-data log-likelihood, we only need to compute the following *sufficient statistics*:

$$\langle \mathbf{X} \rangle = \mathbf{M}^{-1}\mathbf{W}^T(\mathbf{T} - \boldsymbol{\mu}\mathbf{e}^T), \quad (19)$$

$$\langle \mathbf{X}\boldsymbol{\Delta}\mathbf{X}^T \rangle = N\sigma^2\mathbf{M}^{-1} + \langle \mathbf{X} \rangle\boldsymbol{\Delta}\langle \mathbf{X} \rangle^T, \quad (20)$$

where $\mathbf{M} = \mathbf{W}^T\mathbf{W} + \sigma^2\mathbf{I}_q$. Note that all these statistics are computed based on the parameter values obtained from the previous iteration.

In the M-step, to maximize the expectation of the complete-data log-likelihood, the parameters $\{\mathbf{W}, \sigma^2\}$ are updated as follows:

$$\widetilde{\mathbf{W}} = \mathbf{H}\mathbf{W}(\sigma^2\mathbf{I}_q + \mathbf{M}^{-1}\mathbf{W}^T\mathbf{H}\mathbf{W})^{-1}, \quad (21)$$

$$\tilde{\sigma}^2 = \frac{\text{tr}(\mathbf{H} - \mathbf{H}\mathbf{W}\mathbf{M}^{-1}\widetilde{\mathbf{W}}^T)}{d}, \quad (22)$$

where \mathbf{H} is defined in (15). Note that we use \mathbf{W} here to denote the old value and $\widetilde{\mathbf{W}}$ for the updated new value.

4.3 Complexity Analysis

Suppose there are δ nonzero elements in Δ . We can see that the computation cost for \mathbf{H} is $O(dN + d\delta)$. In many applications δ is typically a constant multiple of N . Hence, we can say that the time complexity for computing \mathbf{H} is $O(dN)$.

For the closed-form solution, we have to invert a $d \times d$ matrix. Hence, the computation cost is $O(dN + d^3)$.

For EM, because d is typically larger than q , we can see that the computation cost is $O(dN + d^2qT)$, where T is the number of EM iterations. If the data are of very high dimensionality, EM will be more efficient than the closed-form solution.

5 Experiments

Although PPCA possesses additional advantages when compared with the original non-probabilistic formulation of PCA, they will get similar DR results when there exist no missing values in the data. If the task is to classify instances in the low-dimensional embedding, the classifiers based on the embedding results of PCA and PPCA are expected to achieve comparable results. Hence, in this paper, we adopt PCA as the baseline to study the performance of PRPCA. For the EM algorithm of PRPCA, we use PCA to initialize \mathbf{W} , σ^2 is initialized to 10^{-6} , and $\gamma = 10^{-6}$. Because the EM algorithm and the closed-form solution achieve similar results, we only report the results of the EM algorithm of PRPCA in the following experiments.

5.1 Data Sets and Evaluation Scheme

We use three data sets to evaluate PRPCA. The first two data sets are Cora [17] and WebKB [9]. We adopt the same strategy as that in [27] to preprocess these two data sets. The third data set is the PoliticalBook data set used in [20].

The Cora data set contains the abstracts and references of research papers from the computer science community. For fair comparison, we adopt the same subset of the data as that in [27] for evaluation. The task is to classify each paper into one of the subfields of *data structures* (DS), *hardware and architecture* (HA), *machine learning* (ML), and *programming languages* (PL). Some information about the Cora data set is summarized in Table 1.

Table 1: Information about the Cora data set.

	#classes	#instances	#features
DS	9	751	6,234
HA	7	400	3,989
ML	7	1,617	8,329
PL	9	1,575	7,949

The WebKB data set contains web pages collected from the web sites of computer science departments of four universities (Cornell, Texas, Washington, and Wisconsin). Each web page is labeled as one of seven categories: *student*, *professor*, *course*, *project*, *staff*, *department*, and *“other”*. According to the semantics of *authoritative pages* and *hub pages* [26], we first preprocess the link structure of this data set as follows: if two web pages are co-linked by or link to another common web page, we add a link between these two pages. Then all the original links are removed. After preprocessing, all the directed links are converted into undirected links. Some information about the WebKB data set is summarized in Table 2.

The PoliticalBook data set contains 105 books, 43 of which are labeled as liberal (positive). A relation between two books means that they are frequently bought together by the same customer. The goal is to decide whether or not a specific book is of liberal political inclination. Each book is described by the words on the Amazon.com page for the book. After preprocessing, each book is represented by a feature vector of length 13,178. The preprocessed data can be downloaded from

Table 2: Information about the WebKB data set.

	#classes	#instances	#features
Cornell	7	827	4,134
Texas	7	814	4,029
Washington	7	1,166	4,165
Wisconsin	6	1,210	4,189

<http://www.statslab.cam.ac.uk/~silva>. This data set is used to demonstrate that PRPCA can also work well on data of very high dimensionality.

We adopt the same strategy as that in [27] to evaluate PRPCA on the Cora and WebKB data sets. More specifically, we adopt 5-fold cross validation to evaluate our method. After the learning process of PRPCA, we adopt (19) to perform DR. Then a linear support vector machine (SVM)⁴ is trained for classification based on the low-dimensional representation, which is the same as the testing procedure for the link-content MF method [27]. The average classification accuracy and standard deviation over five repeats are used to report the performance.

For the PoliticalBook data set, we first use PRPCA to perform DR. Then, a Gaussian process classifier [19] is trained based on the low-dimensional representation, which is the same as the testing procedure of the latent Wishart process (LWP) model [16].

5.2 Convergence Speed of EM

We use the DS and Cornell data sets to illustrate the convergence speed of the EM learning procedure of PRPCA. The performance on other data sets has similar characteristics, which is omitted here. With $q = 50$, the average classification accuracy based on 5-fold cross validation against the number of EM iterations T is shown in Figure 2. We can see that PRPCA achieves very promising and stable performance after a very small number of iterations. We set $T = 5$ in all our following experiments.

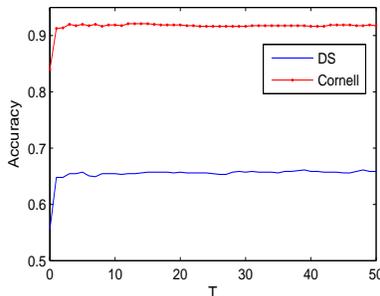


Figure 2: Convergence speed of the EM learning procedure of PRPCA.

5.3 Visualization

We use the PoliticalBook data set to visualize the DR results of PCA and PRPCA. For the sake of visualization, q is set to 2. The results are depicted in Figure 3. We can see that it is not easy to separate the two classes in the latent space of PCA. However, the two classes are better separated from each other in the latent space of PRPCA. Hence, good classification performance can be expected when the examples are classified in the latent space of PRPCA, which will be verified by our subsequent experiment.

5.4 Performance

The dimensionality of Cora and WebKB is moderately high, but the dimensionality of PoliticalBook is very high. We evaluate PRPCA on these two different kinds of data to verify its effectiveness in general settings.

⁴We use LIBSVM [3] to train all the SVM classifiers.

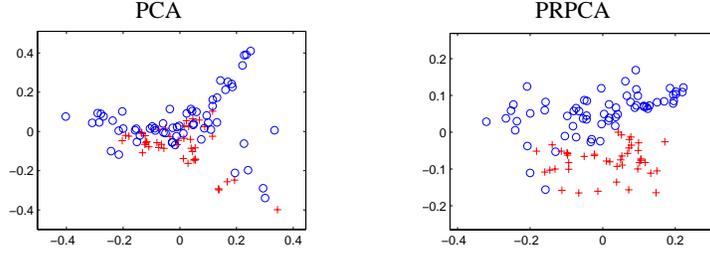


Figure 3: Visualization of data points in the latent spaces of PCA and PRPCA for the PoliticalBook data set. The positive and negative examples are shown as red crosses and blue circles, respectively.

5.4.1 Performance on Cora and WebKB

We perform comparison between PRPCA and PCA to demonstrate that PRPCA can exploit relational information successfully to outperform PCA which discards the relational information. The average classification accuracy with its standard deviation based on 5-fold cross validation against the dimensionality of the latent space q is shown in Figure 4. We can find that PRPCA can dramatically outperform PCA on all the data sets under any dimensionality, which confirms that the relational information is very informative and PRPCA can utilize it very effectively.

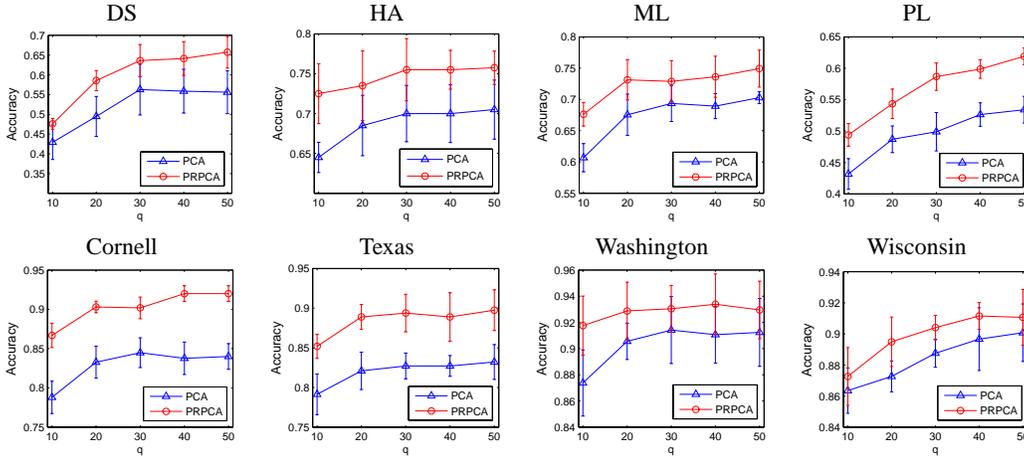


Figure 4: Comparison between PRPCA and PCA on Cora and WebKB.

We also perform comparison between PRPCA and those methods evaluated in [27]. The methods include: *SVM on content*, which ignores the link structure in the data and applies SVM only on the content information in the original bag-of-words representation; *SVM on links*, which ignores the content information and treats the links as features, i.e., the i th feature is *link-to-page*; *SVM on link-content*, in which the content features and link features of the two methods above are combined to give the feature representation; *directed graph regularization (DGR)*, which is introduced in [26]; *PLSI+PHITS*, which is described in [8]; *link-content MF*, which is the joint link-content matrix factorization (MF) method in [27]⁵. As in the link-content MF method, we set $q = 50$ for PRPCA. The results are shown in Figure 5. We can see that PRPCA and link-content MF achieve the best performance among all the evaluated methods. Compared with link-content MF, PRPCA performs slightly better on DS and HA while performing slightly worse on ML and Texas, and achieves comparable performance on the other data sets. We can conclude that the overall performance of PRPCA is comparable with that of link-content MF. Unlike link-content MF which is transductive in nature, PRPCA naturally supports inductive inference. More specifically, we can apply the learned transformation matrix of PRPCA to perform DR for unseen test data, while link-content MF can only perform DR for those data available during the training phase. Very recently, another method proposed by us, called *relation regularized matrix factorization (RRMF)* [15], has achieved better performance than PRPCA on the Cora data set. However, similar to link-content MF, RRMF cannot be used for inductive inference either.

⁵*Link-content sup. MF* in [27] is not adopted here for comparison. Because during the DR procedure link-content sup. MF employs additional label information which is not employed by other DR methods, it is unfair to directly compare it with other methods.

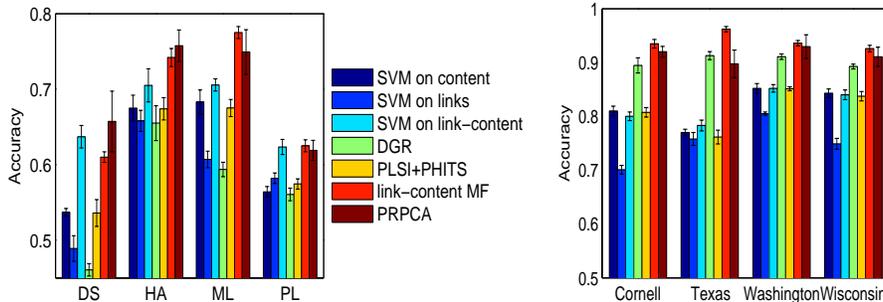


Figure 5: Comparison between PRPCA and other methods on Cora and WebKB.

5.4.2 Performance on PoliticalBook

As in mixed graph Gaussian process (XGP) [20] and LWP [16], we randomly choose half of the whole data for training and the rest for testing. This subsampling process is repeated for 100 rounds and the average *area under the ROC curve* (AUC) with its standard deviation is reported in Table 3, where GPC is a Gaussian process classifier trained on the original feature representation, and relational Gaussian process (RGP) is the method in [6]. For PCA and PRPCA, we first use them to perform DR, and then a Gaussian process classifier is trained based on the low-dimensional representation. Here, we set $q = 5$ for both PCA and PRPCA. We can see that on this data set, PRPCA also dramatically outperforms PCA and achieves performance comparable with the state of the art. Note that RGP and XGP cannot learn a low-dimensional embedding for the instances. Although LWP can also learn a low-dimensional embedding for the instances, the computation cost to obtain a low-dimensional embedding for a test instance is $O(N^3)$ because it has to invert the kernel matrix defined on the training data.

Table 3: Performance on the PoliticalBook data set. Results for GPC, RGP and XGP are taken from [20] where the standard deviation is not reported.

GPC	RGP	XGP	LWP	PCA	PRPCA
0.92	0.98	0.98	0.98 ± 0.02	0.92 ± 0.03	0.98 ± 0.02

6 Conclusion and Future Work

The i.i.d. assumption adopted by most traditional DR methods is unreasonable for relational data. In this paper, we propose a novel DR method for relational data analysis. Although we no longer make the i.i.d. assumption in PRPCA, the learning algorithms for PRPCA can still be devised easily like those for PPCA. Extensive experiments on some real-world data sets show that PRPCA can effectively utilize the relational information to dramatically outperform PCA and achieve state-of-the-art performance.

Extending PRPCA to handle data with missing values and the Bayesian extension of PRPCA will be pursued in our future work. Furthermore, we will also apply PRPCA to model other kinds of networks, such as biological networks [23] and social networks [5].

Acknowledgments

Li and Yeung are supported by General Research Fund 621407 from the Research Grants Council of Hong Kong. Zhang is supported in part by 973 Program (Project No. 2010CB327903). We thank Yu Zhang for some useful comments.

References

- [1] D. J. Bartholomew and M. Knott. *Latent Variable Models and Factor Analysis*. Kendall’s Library of Statistics, 7, second edition, 1999.
- [2] C. M. Bishop. Bayesian PCA. In *NIPS 11*, 1998.

- [3] C.-C. Chang and C.-J. Lin. *LIBSVM: a Library for Support Vector Machines*, 2001. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [4] J. Chang and D. M. Blei. Relational topic models for document networks. In *AISTATS*, 2009.
- [5] J. Chang, J. L. Boyd-Graber, and D. M. Blei. Connections between the lines: augmenting social networks with text. In *KDD*, pages 169–178, 2009.
- [6] W. Chu, V. Sindhwani, Z. Ghahramani, and S. S. Keerthi. Relational learning with Gaussian processes. In *NIPS 19*, 2007.
- [7] F. Chung. *Spectral Graph Theory*. Number 92 in Regional Conference Series in Mathematics. American Mathematical Society, 1997.
- [8] D. A. Cohn and T. Hofmann. The missing link - a probabilistic model of document content and hypertext connectivity. In *NIPS 13*, 2000.
- [9] M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. M. Mitchell, K. Nigam, and S. Slattery. Learning to extract symbolic knowledge from the world wide web. In *AAAI/IAAI*, pages 509–516, 1998.
- [10] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.
- [11] L. Getoor and B. Taskar. *Introduction to Statistical Relational Learning*. The MIT Press, 2007.
- [12] A. K. Gupta and D. K. Nagar. *Matrix Variate Distributions*. Chapman & Hall/CRC, 2000.
- [13] H. Howard. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 27:417–441, 1933.
- [14] I. T. Jolliffe. *Principal Component Analysis*. Springer, second edition, 2002.
- [15] W.-J. Li and D.-Y. Yeung. Relation regularized matrix factorization. In *IJCAI*, 2009.
- [16] W.-J. Li, Z. Zhang, and D.-Y. Yeung. Latent Wishart processes for relational kernel learning. In *AISTATS*, pages 336–343, 2009.
- [17] A. McCallum, K. Nigam, J. Rennie, and K. Seymore. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3(2):127–163, 2000.
- [18] R. Nallapati, A. Ahmed, E. P. Xing, and W. W. Cohen. Joint latent topic models for text and citations. In *KDD*, pages 542–550, 2008.
- [19] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.
- [20] R. Silva, W. Chu, and Z. Ghahramani. Hidden common cause relations in relational learning. In *NIPS 20*. 2008.
- [21] B. Taskar, P. Abbeel, and D. Koller. Discriminative probabilistic models for relational data. In *UAI*, pages 485–492, 2002.
- [22] M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal Of The Royal Statistical Society Series B*, 61(3):611–622, 1999.
- [23] J.-P. Vert. Reconstruction of biological networks by supervised machine learning approaches. In *Elements of Computational Systems Biology*, 2009.
- [24] T. Yang, R. Jin, Y. Chi, and S. Zhu. A Bayesian framework for community detection integrating content and link. In *UAI*, 2009.
- [25] T. Yang, R. Jin, Y. Chi, and S. Zhu. Combining link and content for community detection: a discriminative approach. In *KDD*, pages 927–936, 2009.
- [26] D. Zhou, B. Schölkopf, and T. Hofmann. Semi-supervised learning on directed graphs. In *NIPS 17*, 2004.
- [27] S. Zhu, K. Yu, Y. Chi, and Y. Gong. Combining content and link for classification using matrix factorization. In *SIGIR*, 2007.

7 Appendix

A Derivation of the EM Algorithm for PRPCA

The complete-data log-likelihood is

$$\begin{aligned} L_c &= \ln p(\mathbf{T}, \mathbf{X}) = \ln p(\mathbf{T} | \mathbf{X}) + \ln p(\mathbf{X}) \\ &= -\frac{Nd}{2} \ln \sigma^2 - \frac{1}{2\sigma^2} \left[\text{tr}[(\mathbf{T} - \boldsymbol{\mu}\mathbf{e}^T)\boldsymbol{\Delta}(\mathbf{T} - \boldsymbol{\mu}\mathbf{e}^T)^T] \right. \\ &\quad \left. - 2\text{tr}[(\mathbf{T} - \boldsymbol{\mu}\mathbf{e}^T)\boldsymbol{\Delta}\mathbf{X}^T\mathbf{W}^T] + \text{tr}(\mathbf{W}^T\mathbf{W}\mathbf{X}\boldsymbol{\Delta}\mathbf{X}^T) \right], \end{aligned} \quad (23)$$

where we have omitted the terms independent of the parameters \mathbf{W} and σ^2 . It is easy to find that \mathbf{X} and $\mathbf{X}\boldsymbol{\Delta}\mathbf{X}^T$ are the complete-data sufficient statistics for \mathbf{W} and σ^2 .

Based on the definition of $p(\mathbf{X})$ in (11) and $p(\mathbf{T} | \mathbf{X})$ in (12), and using some properties of matrix variate normal distributions [12], we can get

$$p(\mathbf{X} | \mathbf{T}) = \mathcal{N}_{q,N}(\mathbf{M}^{-1}\mathbf{W}^T(\mathbf{T} - \boldsymbol{\mu}\mathbf{e}^T), \sigma^2\mathbf{M}^{-1} \otimes \boldsymbol{\Phi}), \quad (24)$$

where $\mathbf{M} = \mathbf{W}^T\mathbf{W} + \sigma^2\mathbf{I}_q$.

Then, we can get

$$\langle \mathbf{X} \rangle = \mathbf{M}^{-1}\mathbf{W}^T(\mathbf{T} - \boldsymbol{\mu}\mathbf{e}^T), \quad (25)$$

$$\langle \mathbf{X}\boldsymbol{\Delta}\mathbf{X}^T \rangle = N\sigma^2\mathbf{M}^{-1} + \langle \mathbf{X} \rangle \boldsymbol{\Delta} \langle \mathbf{X} \rangle^T, \quad (26)$$

where the result of $\langle \mathbf{X}\boldsymbol{\Delta}\mathbf{X}^T \rangle$ is derived based on Theorem 3 and the fact that $\text{tr}(\boldsymbol{\Delta}\boldsymbol{\Phi}) = \text{tr}(\mathbf{I}_N) = N$.

The E-step computes the expectation of L_c with respect to $p(\mathbf{X} | \mathbf{T}, \mathbf{W}(t), \sigma^2(t))$, giving

$$\begin{aligned} Q(\mathbf{W}, \sigma^2 | \mathbf{W}(t), \sigma^2(t)) &= \\ &= -\frac{Nd}{2} \ln \sigma^2 - \frac{1}{2\sigma^2} \left[\text{tr}[(\mathbf{T} - \boldsymbol{\mu}\mathbf{e}^T)\boldsymbol{\Delta}(\mathbf{T} - \boldsymbol{\mu}\mathbf{e}^T)^T] \right. \\ &\quad \left. - 2\text{tr}[(\mathbf{T} - \boldsymbol{\mu}\mathbf{e}^T)\boldsymbol{\Delta}\langle \mathbf{X} \rangle^T\mathbf{W}^T] + \text{tr}(\mathbf{W}^T\mathbf{W}\langle \mathbf{X}\boldsymbol{\Delta}\mathbf{X}^T \rangle) \right], \end{aligned} \quad (27)$$

where $\mathbf{W}(t)$ and $\sigma^2(t)$ denote the parameter values of the t th iteration, and $\langle \mathbf{X} \rangle$ and $\langle \mathbf{X}\boldsymbol{\Delta}\mathbf{X}^T \rangle$ are computed based on $\mathbf{W}(t)$ and $\sigma^2(t)$.

In the M-step, $Q(\mathbf{W}, \sigma^2 | \mathbf{W}(t), \sigma^2(t))$ is maximized with respect to \mathbf{W} and σ^2 , giving their new values:

$$\begin{aligned} \mathbf{W}(t+1) &= (\mathbf{T} - \boldsymbol{\mu}\mathbf{e}^T)\boldsymbol{\Delta}\langle \mathbf{X} \rangle^T \langle \mathbf{X}\boldsymbol{\Delta}\mathbf{X}^T \rangle^{-1} = \mathbf{H}\mathbf{W}(t) \left(\sigma^2(t)\mathbf{I}_q + \mathbf{M}^{-1}(t)\mathbf{W}^T(t)\mathbf{H}\mathbf{W}(t) \right)^{-1}, \\ \sigma^2(t+1) &= \frac{\text{tr}(\mathbf{H} - \mathbf{H}\mathbf{W}(t)\mathbf{M}^{-1}(t)\mathbf{W}^T(t+1))}{d}, \end{aligned}$$

where \mathbf{H} is defined in (15). Note that in (21) and (22), for notational convenience, \mathbf{W} and σ^2 are used to denote the old values of the t th iteration, and $\tilde{\mathbf{W}}$ and $\tilde{\sigma}^2$ are for the updated new values of the $(t+1)$ th iteration.