

ISSP 3535 Midterm Project

Branislav Kveton
Intelligent Systems Program
University of Pittsburgh
bkveton@cs.pitt.edu

Mihai Rotaru
Department of Computer Science
University of Pittsburgh
mrotaru@cs.pitt.edu

Introduction

At the end of twentieth century, huge data sets and large distributed environments posed a new challenge for machine learning algorithms. To describe the structure of data more compactly, people started to seek for meaningful components that would characterize the data in a significantly lower dimension.

Probably the best known method for finding characteristic features of data is principal component analysis (PCA). Recently, a probabilistic variation of this approach, probabilistic PCA (PPCA), was proposed, and extended the original method beyond its limits. Better understanding of the probability model that underlies PCA pointed out to the disadvantages of this approach, especially if the modeled problem cannot be characterized by a Gaussian distribution. This effort led into introduction of new models, as PHITS. In PHITS, the underlying distribution is multinomial, which characterizes better problems that deal with counts.

This report focuses on the description and comparison of three major methods that are used for finding representative components in data sets: PCA, PPCA, and PHITS. There is a separate section devoted to every method, and each of them discusses: introduction to the approach, mathematical background, comparison of the method, and experimental results. Experimental results are obtained on two types of data sets: link structures and microarray gene expressions. Preprocessing and analysis techniques used for the datasets are postponed to the section "Preprocessing and Analysis Techniques".

Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a standard technique for dimension reduction and the compression of data. The technique reduces the dimension of the original data k by producing a set of q principal vectors, which are perpendicular and represent the components of the data preserving the highest variance. The simplicity of the reduction and an easily-performed linear transformation between the subspaces contribute

Copyright © 2003, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

to the widespread use of the method. In addition to the solid underlying mathematical theory, there exist a lot of empirical evidence that point out to the scalability of the approach in several fields: authoritative sources in a hyperlinked environment (Kleinberg 1999; Borodin *et al.* 2001), information retrieval (Berry, Drmac, & Jessup 1999), spectral analysis of data, etc.

Introduction to PCA

Let $\mathbf{Y} = (\mathbf{y}_1, \dots, \mathbf{y}_n)$ be a column matrix of n data samples, each of size k . An input vector \mathbf{y} can be expressed by an arbitrary set of perpendicular basis vectors $\mathbf{u}_1, \dots, \mathbf{u}_k$ as

$$\mathbf{y} = \sum_{j=1}^k z_j \mathbf{u}_j,$$

where

$$\mathbf{z}_j = \mathbf{u}_j^T \mathbf{y}.$$

An intuition behind PCA is to approximate the input vector \mathbf{y} by

$$\tilde{\mathbf{y}} = \sum_{j=1}^q z_j \mathbf{u}_j + \sum_{j=q+1}^k b_j \mathbf{u}_j,$$

such that the first q most important vectors \mathbf{u}_j are preserved, while the rest of the vectors is discarded by keeping b_j constant.

The search for these vectors with regards to the mean squared reconstruction error

$$\sum_{i=1}^n \|\mathbf{y}_i - \tilde{\mathbf{y}}\|^2 = \sum_{i=1}^n \sum_{j=q+1}^k (\mathbf{u}_j^T \mathbf{y}_i - b_j)^2$$

leads to the conclusion that these vectors are represented by q eigenvectors with the largest eigenvalues of the matrix $\mathbf{S}\mathbf{S}^T$, where

$$\mathbf{S} = \frac{1}{n} \sum_{i=1}^n (\mathbf{y}_i - \bar{\mathbf{y}})(\mathbf{y}_i - \bar{\mathbf{y}})^T.$$

After the vectors are computed, any input vector \mathbf{y} has a corresponding q dimensional representation $(\mathbf{u}_1^T \mathbf{y}, \dots, \mathbf{u}_q^T \mathbf{y})$, which can be computed in time $O(kq)$.

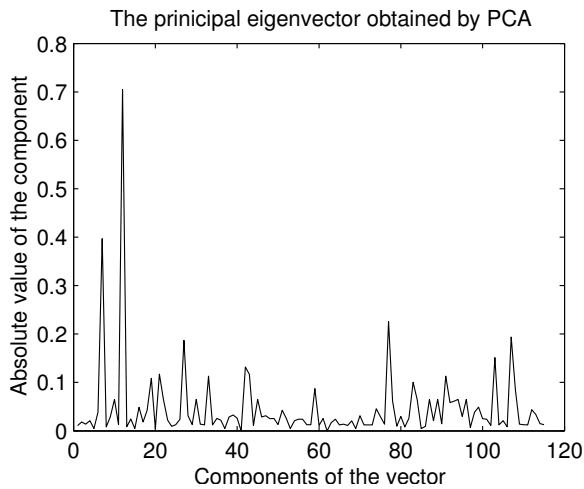


Figure 1: Graphical representation of the principal eigenvector computed by PCA (co-publication data set).

Author	Author number	Score
a_moore	12	0.705
a_gray	7	0.397
r_castano	77	0.225
t_mann	107	0.194
e_mjolsness	27	0.187
t_estlin	103	0.151
j_roden	42	0.132
d_cohn	21	0.117
j_schneider	43	0.116
s_ur	91	0.113

Figure 2: The 10 most representative authors computed by PCA (co-publication data set). The authors are scored according to the absolute value of a particular component in the principal eigenvector.

Experimental Results

The link data set was analyzed with the respect to the principal eigenvector. The absolute value of its components is plotted in Figure 1. The most representative authors according to the principal eigenvector are shown in Figure 2.

The microarray data set was analyzed with the respect to the principal eigenvector. The time to find the eigenvector by PCA was slightly higher than 20 seconds. The absolute values of the eigenvector's components are plotted in Figure 3. Dark squares correspond to the genes's activities that are highly correlated when the phase of fermentation turns into respiration.

Probabilistic Principal Component Analysis (PPCA)

Probabilistic Principal Component Analysis (PPCA) (Tipping & Bishop 1997) is PCA algorithm that emerged as a special case of latent variable model. As

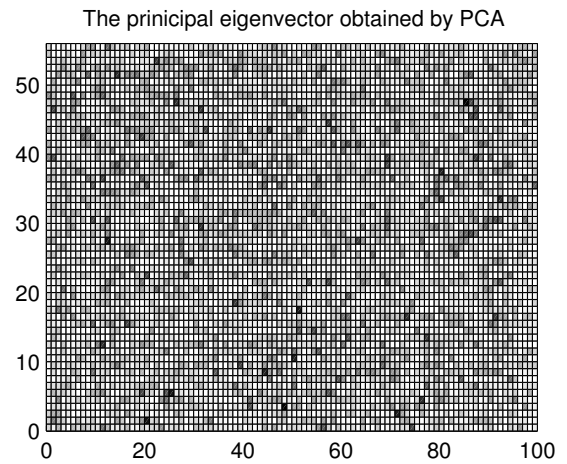


Figure 3: Graphical representation of the principal eigenvector computed by PCA (microarray data set). The original vector of 5602 attributes is due to presentation purposed reduced to the matrix of 56 rows and 100 columns (5600 attributes total). Darker squares correspond to the components of the vector that have high absolute values, as opposing to white squares, which represent close to zero coefficients.

opposing to the original algorithm to solve PCA, this approach is purely probabilistic, which comes with the advantages in terms of computation time and generalization of the model.

PCA as a Latent Variable Model

Latent variable model is a model that seeks an explanation of k dimensional data using a linear projection from a lower q dimensional space. The model can be described by the equation

$$\begin{aligned} \mathbf{y} &= \mathbf{C}\mathbf{x} + \mathbf{v} \\ \mathbf{x} &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \\ \mathbf{v} &\sim \mathcal{N}(\mathbf{0}, \mathbf{R}), \end{aligned}$$

where \mathbf{y} is a centered vector of observation, \mathbf{x} is vector of q independent normally distributed hidden factors, \mathbf{C} is a matrix of the linear projection from the subspace \mathbf{x} to the space \mathbf{y} , and the component \mathbf{v} introduces normally distributed noise in the form of covariance matrix \mathbf{R} . PCA can be viewed as a limiting case of this latent variable model when $\mathbf{R} = \lim_{\epsilon \rightarrow 0} \epsilon \mathbf{I}$. This is the missing link between PCA and graphical models pointed out at the same time by Roweis (Roweis 1998) and Tipping (Tipping & Bishop 1997).

To estimate parameters of the graphical model representation for PCA, also denoted as probabilistic PCA (PPCA), any algorithm that maximizes the likelihood of \mathbf{Y} can be used. For example, Roweis (Roweis 1998) proposed an effective EM algorithm of the form

$$\begin{aligned} \text{e-step:} \quad \mathbf{X} &= (\mathbf{C}^T \mathbf{C})^{-1} \mathbf{C}^T \mathbf{Y} \\ \text{m-step:} \quad \mathbf{C} &= \mathbf{Y} \mathbf{X}^T (\mathbf{X} \mathbf{X}^T)^{-1}. \end{aligned}$$

The whole EM algorithm degrades into a sequence of matrix operation, and an iterative update of two matrices \mathbf{X} and \mathbf{C} is performed until convergence is reached.

Since the proposed algorithm maximizes the likelihood of \mathbf{Y} , it exhibits standard properties of the EM algorithms, and converges to a local maximum of a likelihood function. Moreover, along with the results of Tipping (Tipping & Bishop 1997), the algorithm actually converges to the global maximum, not depending on the starting choice of \mathbf{C} .

PPCA Versus PCA

Together with the new procedure for solving PCA, several advantages and concerns arise. They are discussed in the following paragraph in the form of a comparison between PPCA and PCA.

First, PPCA can handle missing data, which is a problem for PCA. To handle the problem of missing data by PCA, unobserved values are either approximated (the mean of observed values), or the samples containing unobserved values are discarded. The latter approach is not suitable for the domains where missing are inherently present and the number of data samples is not sufficient enough. A good example of such domain is microarray gene analysis.

Second, the computational complexity of PPCA per iteration is only $O(nkq)$, while the computation of the sample covariance matrix in PCA takes $O(nk^2)$ (Roweis 1998). This result is especially encouraging as the input data dimension k is usually huge, while we usually look for a small set of explanations $q \ll k$. On the other hand, one of the concerns for PPCA is that the number of iterations for the algorithm to converge depends on k or n . An empirical study on this problem for a specific dataset was done by (Roweis 1998). He showed that the number of iterations needed for the computation of the principal eigenvector stayed almost the same while the dimension of the input data k varied from several attributes up to 450.

Third, PPCA models can be effectively combined into a mixture of PPCA models (Tipping & Bishop 1999). Even if a number of approaches to solve a mixture of PCA models was proposed, they are usually implemented in two phases: clustering of the input data, and running local PCA for a particular cluster. The disadvantage is the two step approach, and one can imagine that both the clustering and local PCAs should be optimized together. This type of problem can be easily handled by introducing a hidden variable, which is responsible for switching between different PCA models.

Finally, vectors obtained by PPCA in the form of the column matrix $\mathbf{C} = (\mathbf{c}_1, \dots, \mathbf{c}_q)$ are not necessarily perpendicular, and thus different from the eigenvectors $\mathbf{u}_1, \dots, \mathbf{u}_q$ computed by PCA. In fact, for $q = 1$, the trivial result $\mathbf{c}_1 = \lambda \mathbf{u}_1$ arises, where λ is a constant. For $q > 1$, the subspace spanned by $\mathbf{c}_1, \dots, \mathbf{c}_q$ is the same as the one spanned by $\mathbf{u}_1, \dots, \mathbf{u}_q$, so both transformations are the same. This fact can be verified by comparing

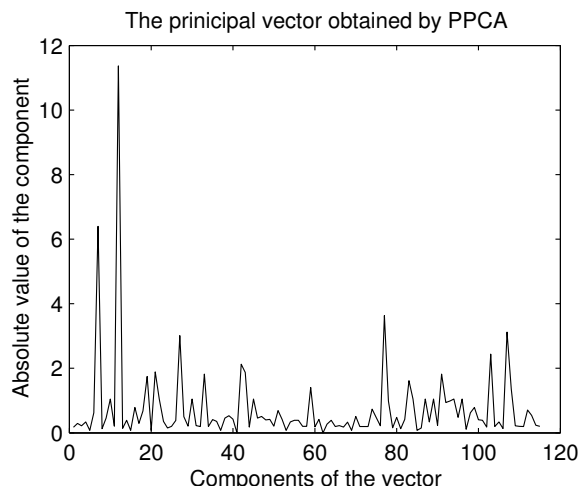


Figure 4: Graphical representation of the principal vector computed by PPCA (co-publication data set).

Author	Author number	Score
a_moore	12	10.478
a_gray	7	5.899
r_castano	77	3.349
t_mann	107	2.878
e_mjolsness	27	2.778
t_estlin	103	2.250
j_roden	42	1.960
d_cohn	21	1.743
j_schneider	43	1.727
s_ur	91	1.678

Figure 5: The 10 most representative authors computed by PPCA (co-publication data set). The authors are scored according to the absolute value of a particular component in the principal vector.

results of the transformations

$$(\mathbf{u}_1, \dots, \mathbf{u}_q)(\mathbf{u}_1, \dots, \mathbf{u}_q)^T(\mathbf{y}_i - \bar{\mathbf{y}}) + \bar{\mathbf{y}}$$

and

$$\mathbf{C}\mathbf{x}_i + \bar{\mathbf{y}}$$

to the original space.

Experimental Results

The link data set was analyzed with the respect to the principal vector. The absolute value of its components is plotted in Figure 4. The most representative authors according to the principal vector are shown in Figure 5. Even if the components of the principal vector are different from those obtained by PCA, the difference is only in a constant multiplying the vector, and hence the interpretation of the results stay the same.

The microarray data set was analyzed with the respect to the principal vector. The time to find the vector by PPCA was well below 1 second, which is significantly better than the one of PCA. The absolute

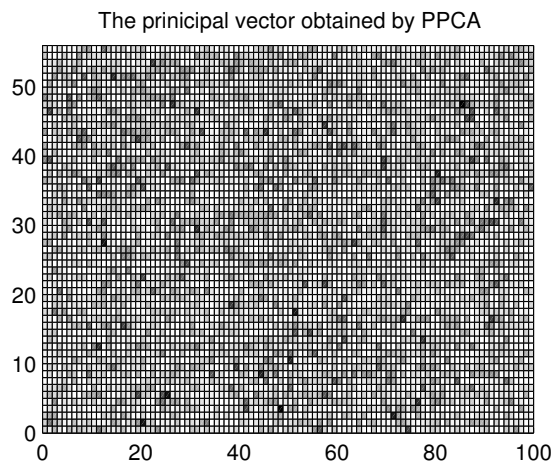


Figure 6: Graphical representation of the principal vector computed by PPCA (microarray data set). The original vector of 5602 attributes is due to presentation purposed reduced to the matrix of 56 rows and 100 columns (5600 attributes total). Darker squares correspond to the components of the principal vector that have high absolute values, as opposing to white squares, which represent close to zero coefficients.

values of the vector’s components are plotted in Figure 6. Even if the vector is different from the one obtained by PCA, the proportion between vector’s components stays the same. Dark squares correspond to the genes’ activities that are highly correlated when the phase of fermentation turns into respiration.

Preprocessing and Analysis Techniques

For the links data set analysis, we have chosen the co-publication data set from the Auton Lab at Carnegie Mellon University (Kubica *et al.* 2003). The data set contains co-author information on 94 papers, and each of them may be co-authored by one of 115 people.

In the context of the link analysis, we are primary interested in the principal eigenvector. This vector represents the most authoritative, in terms of published papers, group of people in the field, which is a parallel between the most authoritative documents analyzed by Cohn (Cohn & Chang 2000). Intuitively, other eigenvectors, ranked according to their eigenvalues, should represent subcommunities that are not captured by the principal eigenvector. Contrary to this intuition, Borodin (Borodin *et al.* 2001) showed for several link structures that the second eigenvector may or may not correspond to other communities. Based on these results, we decided to analyze only the principal vector.

For the microarray gene expression analysis, we have chosen a DNA microarray containing genes of *Saccharomyces cerevisiae* (DeRisi, Iyer, & Brown 1997) obtained from (Gollub *et al.* 2003). The genes are monitored over a time span of 20 hours while the organism

shifts from the stage of fermentation into respiration. The value that is analyzed, and hopefully indicative of the gene’s activity, is \log_2 of R/G normalized ratio (mean). The data set contains only 7 measurements and more than 5000 attributes, but there are no results for multiple experiments as in other data sets, which makes the preprocessing and analysis easier. On the other hand, the small number of samples and the large number of attributes pose a real challenge for dimension reduction techniques. To assure fair comparison between PCA and PPCA, missing values are estimated as the mean of observed data points.

In the context of the microarray analysis, we are primary interested in the principal eigenvector, which represents the most correlated change of genes’ activities in the transition between the stages of fermentation and respiration. Consequentially, the j th gene undergoes a high change in its activity, if the absolute value of the j th component of the principal eigenvector is high comparing to the other components. On the other hand, if the absolute value of the j th component is low comparing to the other components, the activity of the j th gene does not change significantly.

Probabilistic HITS (PHITS)

Analytic decomposition techniques like PCA and LSA have been successfully applied in the analysis of link structures like citations networks, document-term co-occurrence). While efficient and simple, these techniques fail to give a theoretical interpretation of their results. For example, in the case of PCA applied to citation networks, eigenvectors are associated with communities, but determining the authority of a document in a certain community relies on an ad-hoc measurement: the size of the document projection on the eigenvector; the higher the document "loading" on the eigenvector, the more authoritative the document is.

Interpreting the result of PCA relies on the assumption that citations are generated by the scientific domain the paper is part of (a paper from Machine Learning domain will most likely generate citations to other papers from Machine Learning than papers from Computational Theory). It is like domains (which are hidden in the citation data) act as citation generators. This is very similar to the probabilistic latent variable models: the data we observe are generated by a set of hidden variables (also known as factors or aspects).

The main idea of latent models approach to link analysis is to break the link between the two ends of the links using a latent variable. Since the number of values for the latent variable is much smaller than the one for observed variables, the latent variable acts as a bottleneck. In the same time the observable variables become independent given the latent variable.

We will briefly describe the probabilistic latent approach for two link structures: word-document co-occurrence and citations.

In the case of document-word co-occurrence problem, LSA is a well known technique that tries to map high

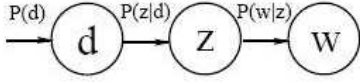


Figure 7:

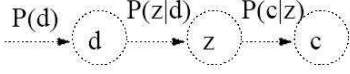


Figure 8:

dimensional term-document count vectors to a smaller space (called the latent semantic space). The hope is that terms with similar meaning will be mapped to the same direction in the latent space. The transformation is computed by doing the SVD decomposition of the term-document co-occurrence matrix. The probabilistic latent approach, PLSA, uses a probability distribution over the documents $P(d)$, then each document is a combination of aspects $P(z|d)$ and, last, each aspect z yields a certain distributions over the words $P(d|c)$ (see Figure 7). The joint encoded in this model acts as non-negative decomposition of the co-occurrence table:

$$P(d, w) = \sum_z P(z)P(d|z)P(w|z)$$

which can be rewritten as

$$P(d, w) = \mathbf{U}\Sigma\mathbf{V}^T$$

where

$$\begin{aligned} \mathbf{U} &= P(d|z) \\ \Sigma &= \text{Diag}(P(z)) \\ \mathbf{V} &= P(w|z) \end{aligned}$$

For the citation data, the analytical approach employed in HITS computes the eigenvector of the co-occurrence matrix $\mathbf{M}^T\mathbf{M}$ and ranks the documents according to the "load" of the document on the principal eigenvector. Since only the largest eigenvector is used, other communities (domains) that might be present in the data are ignored, thus decreasing the performance in finding authoritative documents. On the other hand, the probabilistic approach tries to estimate the distributions over a given number of domains (the number is given but not the domains). The graphical model it is identical with the one for PLSA: it puts the document w.r.t. citing on top and the document w.r.t. to being cited at the bottom (see Figure 8). Defining an authoritative document in this framework is simple: find the document c with the highest value of conditional $P(c|z)$. Moreover, other interesting inferences can be deducted from this model that PCA like approaches do not offer. For example, one can inquire about the community membership of a document by inspecting

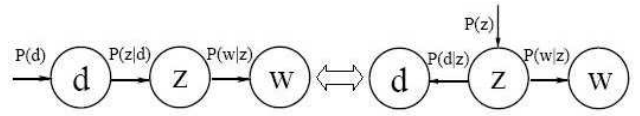


Figure 9:

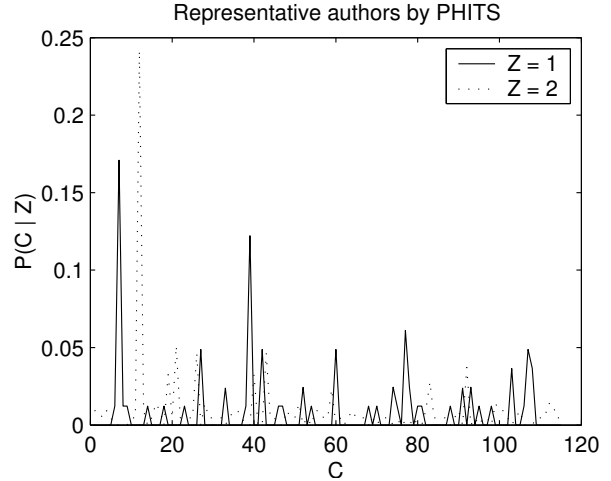


Figure 10: Graphical representation of the representative authors computed by PHITS with the latent variable Z of size 2 (co-publication data set).

$P(z|c)$ or the authority of documents from a domain into another domain.

Learning the models in both cases relies on the fact that the structure is identical with the multinomial Naïve Bayes model with hidden class (see Figure 9). Simple calculation can relate the formulas for updating of parameters from the class slides to the ones presented in both papers. To avoid overfitting and local maxima, a tempering version of EM is introduced by (Hofmann 1999).

Experimental Results

The link data set was analyzed for two different sizes of the latent variable Z : 2 and 4.

Figure 10 shows the distribution of authors over the latent variable Z of size 2, which immediately points out a distinction between the results obtained by PCA and PHITS. The two highest ranked authors by the principal eigenvector, "a_moore" (number 12) and "a_gray" (number 7), belong to two different components of Z , which contradicts the notion of being in the same eigenvector. Deeper inspection of the data set shows that PHITS model is indeed true. Even if "a_moore" published 52 papers, and "a_gray" published 14, there are only two papers that they wrote together. This link is not significant enough as both authors mostly cooperated with other people.

Figure 11 shows the distribution of authors over the latent variable Z of size 4. The distribution is differ-

ent from the one for the latent variable of size 2, and show that the size of the latent variable is an important parameter of the model. If the size of Z is small, the variable may not catch the subgroups of authors well. On the other hand, if the size of Z is bigger, the model may generalize poorly.

Conclusion

Probabilistic latent approaches to link analysis have both advantages and disadvantages. The most important advantage is that it gives a clear probabilistic interpretation of the result. Moreover, one can combine such probabilistic models to build a unified probabilistic model that can model the data even better by capturing semantics along multiple types of links ((Cohn & Hofmann 2001) combined words, documents and citations using 3-way factoring). Computationally, the probabilistic approaches are comparable and sometimes faster.

The major drawback of these methods is the fact that the number of factors has to be set a priori. Choosing the right number of factors has great influence on the model as described in our results section.

References

- Berry, M. W.; Drmac, Z.; and Jessup, E. R. 1999. Matrices, vector spaces, and information retrieval. *SIAM* 41(2):335–362.
- Borodin, A.; Roberts, G. O.; Rosenthal, J. S.; and Tsaparas, P. 2001. Finding authorities and hubs from link structures on the world wide web. In *World Wide Web*, 415–429.
- Cohn, D., and Chang, H. 2000. Learning to probabilistically identify authoritative documents. In *Proc. 17th International Conf. on Machine Learning*, 167–174. Morgan Kaufmann, San Francisco, CA.
- Cohn, D., and Hofmann, T. 2001. The missing link - a probabilistic model of document content and hypertext connectivity. In *Neural Information Processing Systems 13*.
- DeRisi, J. L.; Iyer, V. R.; and Brown, P. O. 1997. Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science* 278(5338):680–686.
- Gollub, J.; Ball, C. A.; Binkley, G.; Demeter, J.; Finkelstein, D. B.; Hebert, J. M.; Hernandez-Boussard, T.; Jin, H.; Kaloper, M.; Matese, J. C.; Schroeder, M.; Brown, P. O.; Botstein, D.; and Sherlock, G. 2003. The stanford microarray database: Data access and quality assessment tools. *Nucleic Acids Res.* 31(1):94–96.
- Hofmann, T. 1999. Probabilistic latent semantic analysis. In *Proc. of Uncertainty in Artificial Intelligence, UAI'99*.
- Kleinberg, J. M. 1999. Authoritative sources in a hyperlinked environment. *Journal of the ACM* 46(5):604–632.

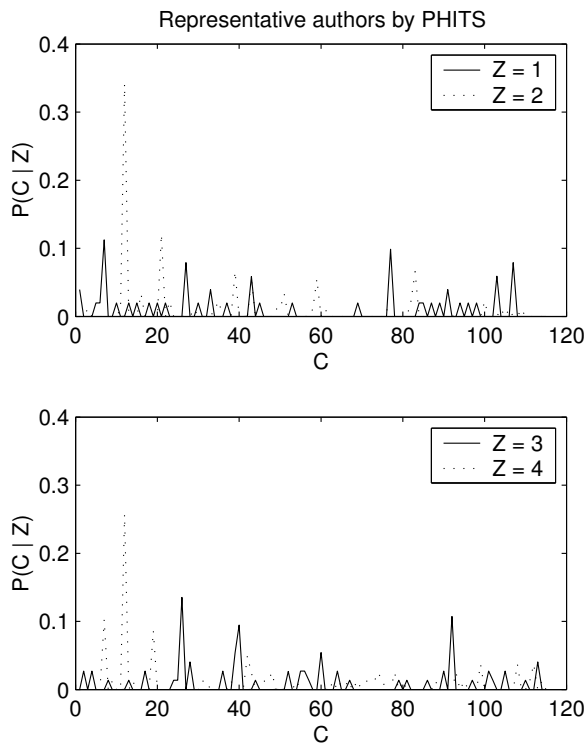


Figure 11: Graphical representation of the representative authors computed by PHITS with the latent variable Z of size 4 (co-publication data set).

Kubica, J.; Moore, A.; Cohn, D.; and Schneider, J. 2003. Finding underlying connections: A fast graph-based method for link analysis and collaboration queries. In Fawcett, T., and Mishra, N., eds., *Proceedings of the 2003 International Conference on Machine Learning*, 392–399. AAAI Press.

Roweis, S. 1998. EM algorithms for PCA and SPCA. In Jordan, M. I.; Kearns, M. J.; and Solla, S. A., eds., *Advances in Neural Information Processing Systems*, volume 10. The MIT Press.

Tipping, M., and Bishop, C. 1997. Probabilistic principal component analysis.

Tipping, M. E., and Bishop, C. M. 1999. Mixtures of probabilistic principal component analysers. *Neural Computation* 11(2):443–482.