Latent block principal component analysis for binary tables

Rodolphe Priam

Latent block principal component analysis for binary tables

R. Priam[*]

March 26, 2021

Abstract

A pca method embedded with a co-clustering for binary tables is proposed via a probabilistic latent variable framework. Two local quadratic approximations of the objective function and a regularization of the parameters lead to iterative expectation maximization algorithms for the inference of the parameters.

Keywords: Bernoulli latent block mixture model, co-clustering, pca, binary matrix, BEM algorithm

1 Introduction

For exploratory data analysis of binary tables, several methods have been proposed in the literature from diverse points of view. We are mainly interested in matricial approaches related to the principal component analysis (pca) [8] for binary tables for reducing the two dimensions of these tables. When the data matrix is defined from the set I of objects (rows, observations) and the set J of variables (columns, attributes), the dimensions are these two sets or their sizes. Reducing the dimensionality is able to show the rows or the columns in a subspace dramatically smaller than the former space, a visualization becomes possible but also a post analysis such as a discriminant one. This is how pca is able to visualize the rows and the columns on the plane thanks to this property. Several linear methods exist in the literature for binary data, such as correspondence analysis [8, 1] or sparse logistic pca [10, 9] and also [19, 16, 11]. Here, a nonlinear version is of main interest because the linearity alone is not able to find the true relations betweens the objects from I or from J in a synthetic way. For improving the limits due to the linear setting, clustering is a powerful choice able to better discover the underlying shape of the data cloud, such was proposed in [20] with a mixture of probabilistic pca. A family of methods has been defined for visualization, it is based on a clustering procedure with vicinity constraints for the clusters. These methods are seen in the literature as a nonlinear pca with a projection on a surface instead of the linear projection on the plane in pca. This family is named self-organizing map, it generalizes a former algorithm called the Kohonen's map[7] which is a k-means method which integrates the constraints at the level of the mean centers. For a sparse binary table, this idea was introduced with an alternative clustering model for co-clustering, the latent block model [5], by a method called herein blockgtm [13] for visualization of binary tables. Generally a co-clustering is dramatically more efficient than a clustering of the two sets separately because the number of parameters is parsimonious and the clustering of one dimension is involved [2, 3, 4] in the clustering of the other dimension. In order to get a reduction method instead of a visualization method, we propose to change the model by adding another set of latent vectors to be estimated. By this way the symmetry of the learning is retrieved as explained next after. In this paper, we present in section 2 the proposed model for binary data with its clustering foundation followed by the parameterization. In Section 3, an algorithm is given for the inference of the parameters with the new constraints. The last section is the concluding part with perspectives.

2 Proposed model and criteria

For principal component analysis, the clusters depend on latent vectors, from the two dimensions (rows and columns) in order to reduce the data table from the two sides. Thus in order to share

[*] rpriam@gmail.com

such reducing purpose, we propose the new method named *blockpca* or *latent block principal component analysis* by rewriting the parameters from the Bernoulli block mixture model with these new latent vectors and adding a penalization for avoiding the empty clusters. The inference of the parameters is based on a generalized expectation maximization (EM) algorithm such as blockgtm. In this section, the model is presented before the inference of the parameters.

2.1 Latent block mixture model

In the following, the $n \times d$ data table is defined by $\mathbf{x} = \{(x_{ij}); i \in I \text{ and } j \in J\}$ where $x_{ij} \in \{0,1\}$. The purpose of a method of block clustering is to partition the table by blocks with a distribution of the cells for each block. In [5, 6], the methods from Govaert [3, 4] are embedded in a probabilistic mixture [17]. The resulting model is called the latent block mixture model (LBM) [5, 6] which is a mixture model which sums over the crossproduct of two sets of assignments $\mathcal{Z} \times \mathcal{W}$. Here $\mathcal{Z}$ and $\mathcal{W}$ denote the sets of all possible assignments $\mathbf{z}$ of I and $\mathbf{w}$ of J. The observed x_{ij} cells are generated by $n \times d$ random variables which are assumed to be independent if $\mathbf{z}$ and $\mathbf{w}$ are known. Let also define $\mathbf{p} = (p_1, \ldots, p_g)$ (resp. $\mathbf{q} = (q_1, \ldots, q_m)$) for the vectors of probabilities p_k (resp. q_ℓ) that a row (resp. a column) belongs to the k^{th} component (resp. ℓ^{th} component). Hence, this leads to the pdf:

$$f(\mathbf{x};\theta) = \sum_{(\mathbf{z},\mathbf{w})} \prod_{i,k} p_k^{z_{ik}} \prod_{j,\ell} p_\ell^{z_{j\ell}} \prod_{i,j,k,\ell} \varphi(x_{ij};\alpha_{k\ell})^{z_{ik}w_{j\ell}} .$$

Here, $\varphi(.;\alpha_{k\ell})$ is the pdf in a block defined for $x_{ij} \in \{0,1\}$ and the unknown parameter $\alpha_{k\ell}$. They are aggregated into the vectorial parameter θ which is written from $\alpha = (\alpha_{11}, \ldots, \alpha_{gm})$ with a vectorial form, $\mathbf{p}$ and $\mathbf{q}$. For binary data the pdf is a Bernoulli law which leads to the Bernoulli latent block model [5]. In a block, the pdf of a cell is defined as follows,

$$\varphi(x_{ij};\alpha_{k\ell}) = (\alpha_{k\ell})^{x_{ij}}(1-\alpha_{k\ell})^{1-x_{ij}} .$$

Next, the parameters $\alpha_{k\ell}$ are written with a function of latent vectors for reduction purpose, and a learning algorithm is given for this new model.

2.2 Parameterization with latent spaces

Herein, the parameters $\alpha_{k\ell}$ of the block mixture model are modeled with two sets of vectors. To keep the dependence on ℓ and k of $\alpha_{k\ell}$, when $h \in \mathbb{N}_+$ is a positive integer, it is introduced inner products defined as: $v_\ell^T u_k$ from the latent variables $\{v_\ell\}$ where $v_\ell \in \mathbb{R}^h$ and $\{u_k\}$ where $u_k \in \mathbb{R}^h$. Each product is mapped into a probability with a sigmoid function $\sigma(.)$:

$$\alpha_{k\ell} = \frac{exp(v_\ell^T u_k)}{1+exp(v_\ell^T u_k)} .$$

This leads to suitable parameters for the block model. With this formulation, the latent parameter α, in a matricial form of size $g \times m$, is replaced by the two latent matrices U and V,

$$\begin{array}{rcl} V & = & [v_1|v_2|\cdots|v_m]^T \\ U & = & [u_1|u_2|\cdots|u_g]^T . \end{array}$$

This model remains parsimonious because h is small in practice.

2.3 Algorithm BEM

In this section, the estimation of the parameter θ is studied via the likelihood maximization of the latent block mixture model. For this purpose an algorithm called BEM [5, 6] generalizes the usual EM algorithm from mixture model for one partition to two partitions (or more) instead of just one. The new completed data is the vector $(\mathbf{x},\mathbf{z},\mathbf{w})$ where the unobservable vectors $\mathbf{z}$ and $\mathbf{w}$ are the row and column labels. It is denoted the posterior probabilities c_{ik} that a data row i is attributed to a row cluster k and $d_{j\ell}$ that a data column j is attributed to a column cluster ℓ. The algorithm has two steps, like EM, which are iterated until convergence.

- **E-step:** The posterior probabilities are found by solving the same problem than in the unconstrained case but with the new parameterization, such that:

$$c_{ik}^{(t)} \propto \prod_\ell(\sigma(v_\ell^T u_k))^{u_{i\ell}}(1-\sigma(v_\ell^T u_k))^{d_\ell - u_{i\ell}} , \quad (1)$$

$$d_{j\ell}^{(t)} \propto \prod_k(\sigma(v_\ell^T u_k))^{v_{jk}}(1-\sigma(v_\ell^T u_k))^{c_k - v_{jk}} . \quad (2)$$

- **M-step:** Here the proportions are supposed not equal. This results into the two following criteria. For the matrix U, let rewrite the objective function as the sum:

$$\begin{aligned} \tilde{Q}(\theta|\theta^{(t)}) &= \textstyle\sum_k \tilde{Q}_k(u_k|\theta^{(t)}) \\ \tilde{Q}_k(u_k|\theta^{(t)}) &= \textstyle\sum_\ell y^{(t)}_{k\ell} v_\ell^T u_k \\ &\quad -c^{(t)}_k d^{(t)}_\ell \log(1+e^{v_\ell^T u_k}) \,. \end{aligned} \tag{3}$$

For the matrix V, let rewrite the objective function as the sum:

$$\begin{aligned} \tilde{Q}(\theta|\theta^{(t)}) &= \textstyle\sum_\ell \tilde{Q}_\ell(v_\ell|\theta^{(t)}) \\ \tilde{Q}_\ell(v_\ell|\theta^{(t)}) &= \textstyle\sum_k y^{(t)}_{k\ell} v_\ell^T u_k \\ &\quad -c^{(t)}_k d^{(t)}_\ell \log(1+e^{v_\ell^T u_k}) \,. \end{aligned} \tag{4}$$

The optimization problem for maximizing the criteria (3) or (4) depending on U and V can be performed also after their corresponding a posteriori probabilities.

Here, it has been denoted the quantities $y^{(t)}_{k\ell} = \sum_{i,j} c^{(t)}_{ik} d^{(t)}_{j\ell} x_{ij}$, $u^{(t)}_{i\ell} = \sum_j d^{(t)}_{j\ell} x_{ij}$, $d^{(t)}_\ell = \sum_j d^{(t)}_{j\ell}$ and $v^{(t)}_{jk} = \sum_i c^{(t)}_{ik} x_{ij}$, $c^{(t)}_k = \sum_i c^{(t)}_{ik}$. A limit with the objective function is the problem of empty clusters which comes with a larger number of blocks than the natural one, which is not suitable with our model. Hence, it is added a penalization with η a constant value in order to insure that the clusters have all different positions in the space,

$$\begin{aligned} \breve{Q}_k(u_k|\theta^{(t)}) &= \tilde{Q}_k(u_k|\theta^{(t)}) + \tfrac{\eta}{2} \textstyle\sum_{k'} ||u_k - u_{k'}||^2 \\ \breve{Q}_\ell(v_\ell|\theta^{(t)}) &= \tilde{Q}_\ell(v_\ell|\theta^{(t)}) + \tfrac{\eta}{2} \textstyle\sum_{\ell'} ||v_\ell - v_{\ell'}||^2 \,. \end{aligned} \tag{5}$$

Note that cancelling η allows to retrieve the former form for the function to optimize, in case of the penalization is not required. Alternative penalizations may be possible but not tested herein, without loss of generality. For the estimation of the parameters, the proposed algorithm is further presented with the analytical expressions of the updates at the M-step next section.

3 M-step and detailed algorithm

An algorithm for the inference of the parameters is presented in this section with first the approximations in stake for a matricial expression of the updates at each iteration, followed by all the steps of the algorithm.

3.1 Two quadratic approximations

At the M-step a nonlinear function is optimized because of the sigmoid functions. The problem to be solved is the following maximization:

$$\theta^{(t+1)} = argmax_\theta \, \breve{Q}(\theta|\theta^{(t)}) \,.$$

For a solution to this problem, a second order approximation of $\breve{Q}$ is presented. The symmetry of the two original mirrored formulas in BEM for each side of the table is not anymore lost on the contrary to blockgtm where only rows were mapped, hence only one side is explained here for V, while the other side for U is found by transposition. Two approaches are considered next, a Newton-Raphson approximation (NR) and a variational approximation.

- **M-step with a 2$^{\text{nd}}$ order approximation:** A NR algorithm is considered by using the first order and second order derivatives of the criterion. This approach is generally able to increase the log-likelihood at each step of EM. There is an interaction between the vectors v_ℓ because of the penalization, hence the problem cannot be solved for each ℓ separately. Without the penalization, the Hessian matrix for the whole set of parameters is block-diagonal $H = diag_\ell(H_\ell)$ with a block H_ℓ for each ℓ. With the penalization, the gradient vectors q_{v_ℓ} are not anymore defined for each v_ℓ independently. It is denoted $C = (c^{(t)}_{ik})$ a $g \times n$ matrix of posterior probabilities, $V = (v^{(t)}_{jk})$ a $g \times d$ matrix of sufficient statistics, $G = (c^{(t)}_k)$ and $F_\ell = (\alpha^{(t)}_{k\ell})$ are $g \times g$ diagonal matrices, $\alpha_\ell = (\alpha^{(t)}_{k\ell})$ a $g \times 1$ vector, $u_\ell = (u^{(t)}_{i\ell})$ a $n \times 1$ vector, $d_\ell = (d^{(t)}_{j\ell})$ a $d \times 1$ vector, and $d_{(\ell)} = \sum_j d^{(t)}_{j\ell}$ is a scalar. Let's $b^{(t)}_\ell$ and $H^{(t)}_\ell$ stand respectively for the first and second order derivatives of $\tilde{Q}_\ell$ at time t. A local quadratic approximation of the objective function in a vicinity of the current solution denoted $v^{(t)}_\ell$ is as follows when keeping the penalization

$$\left[diag \left\{ \begin{pmatrix} H_1^{(t)} - 2\eta\mathbb{I}_h \\ H_2^{(t)} - 2\eta\mathbb{I}_h \\ \vdots \\ H_m^{(t)} - 2\eta\mathbb{I}_h \end{pmatrix} \right\} + \eta \begin{pmatrix} \mathbb{I}_h & \mathbb{I}_h & \cdots & \mathbb{I}_h \\ \mathbb{I}_h & \mathbb{I}_h & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbb{I}_h \\ \mathbb{I}_h & \dots & \mathbb{I}_h & \mathbb{I}_h \end{pmatrix} \right] \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{pmatrix} = \begin{pmatrix} b_1^{(t)} - H_1^{(t)} \bar{v}_1^{(t)} \\ b_2^{(t)} - H_2^{(t)} \bar{v}_2^{(t)} \\ \vdots \\ b_m^{(t)} - H_m^{(t)} \bar{v}_m^{(t)} \end{pmatrix}. \qquad (6)$$

Figure 1: Linear regression for the M-step.

which is already quadratic:

$$\begin{array}{rl} & \breve{Q}_\ell(v_\ell|\theta^{(t)}) \\ \approx & \tilde{Q}_\ell(v_\ell^{(t)}|\theta^{(t)}) + (v_\ell - \bar{v}_\ell^{(t)})^T b_\ell^{(t)} \\ + & \frac{1}{2}(v_\ell - \bar{v}_\ell^{(t)})^T H_\ell^{(t)} (v_\ell - \bar{v}_\ell^{(t)}) \\ + & \frac{\eta}{2} \sum_{\ell'} ||v_\ell - v_{\ell'}||^2 . \end{array} \qquad (7)$$

The NR algorithm performs the maximization of a local quadratic approximation instead of the former objective function. The matrix and the vectors are as follows:

$$\begin{array}{rcl} H_\ell^{(t)} & = & -d_{(\ell)} U^T G F_\ell (\mathbb{I}_h - F_\ell) U \\ b_\ell^{(t)} & = & U^T \left(y_\ell - d_{(\ell)} F_\ell g_k \right) \\ \bar{v}_\ell^{(t)} & = & v_\ell^{(t)} . \end{array} \qquad (8)$$

Note that the dimensions of the matrices H_ℓ remain small even when the data matrix is large because only the size h is involved. Another way to perform the optimization is to approximate the sigmoid function with a quadratic function as explained next.

- **M-step with a variational bound:** For a variational approach (VR), the function φ may be approximated for instance. In this case, a bound on the sigmoid function [15] might be relevant. By convexity, this is written as a upper bound of $\sigma(a)$ as an exponential of a quadratic function of a and ε multiplied with $\sigma(\varepsilon)$ where ε is an unknown parameter to be computed. Here a is a scalar in $\mathbb{R}$, while ε is the variational parameter, and $\lambda(\varepsilon) = \frac{1}{4\varepsilon} tanh\left(\frac{\varepsilon}{2}\right)$. As previously the objective function cannot break into m different criteria which can be maximized independently. Let's denote $\varepsilon_\ell = (\varepsilon_{1\ell}, \varepsilon_{2\ell}, \cdots, \varepsilon_{g\ell})^T$ where at the iteration time each component is computed $\varepsilon_{k\ell}^{(t)} = |(u_k^{(t)})^T v_\ell^{(t)}|$. Then, by using the bound on the sigmoid, the criterion to optimize may be written as follows:

$$\begin{array}{rl} & \breve{Q}_\ell(v_\ell|\theta^{(t)}) \\ \geq & +(v_\ell - \bar{v}_\ell^{(t)})^T \tilde{b}_\ell^{(t)} \\ & -\frac{1}{2}(v_\ell - \bar{v}_\ell^{(t)})^T \tilde{H}_\ell^{(t)} (v_\ell - \bar{v}_\ell^{(t)}) \\ & +d_\ell^{(t)} \sum_k c_k^{(t)} \left\{ \log \sigma(\varepsilon_{k\ell}) \right\} \\ - & d_{(\ell)}^{(t)} \sum_k c_k^{(t)} \left\{ \lambda(\varepsilon_{k\ell})(-\varepsilon_{k\ell}^2) + \frac{1}{2}\varepsilon_{k\ell} \right\} \\ + & \frac{\eta}{2} \sum_{\ell'} ||v_\ell - v_{\ell'}||^2 \\ = & \breve{Q}_\ell(v_\ell, \varepsilon_\ell | v_\ell^{(t)}, \varepsilon_\ell^{(t)}) . \end{array} \qquad (9)$$

Here, g_k denotes the column vector compound of the elements $\sum_i c_{ik}^{(t)}$ and y_ℓ is the column vector compound of the elements $y_{k\ell}^{(t)}$. Let's also have Λ_ℓ for the diagonal matrix with diagonal elements equal to $\lambda_{k\ell} = \lambda(\varepsilon_{k\ell})$. The expression is as for the NR procedure with new corresponding quantities:

$$\begin{array}{rcl} \tilde{b}_\ell^{(t)} & = & U^T \left(y_\ell - \frac{d_{(\ell)}}{2} g_k \right) \\ \tilde{H}_\ell^{(t)} & = & 2 d_{(\ell)} U^T G \Lambda_\ell U \\ \bar{v}_\ell^{(t)} & = & 0_h . \end{array} \qquad (10)$$

As for NR it is obtained a new maximization problem without a multidimensional nonlinear function involved but instead several quadratic problems which can be performed analytically. The dimensionality of the problem is only of order h instead of the size of the table.

3.2 Algorithm for learning the parameters

The two algorithms lead to the same form of quadratic approximation, such as finally, the gradient w.r.t. $\mathbf{v} = vect(V) = (v_1^T, \cdots, v_m^T)^T$ is written by the derivative of the criterion. The two version of the M-step (NR and VR) have been unified under a similar notation and a similar regularization.

```
- Initialization:
Initialize $\{c_{ik}^{(0)}\}$, $\{d_{j\ell}^{(0)}\}$, $U^{(0)}$, $V^{(0)}$.

- E-Step:
Update $\{c_{ik}^{(t)}\}$ by (1), and $\{d_{j\ell}^{(t)}\}$ by (2), alternatively.

- M-Step:
Update $\{\varepsilon_k\}$,$\{\varepsilon_\ell\}$ if needed and $U$,$V$ from (6),

  $V \leftarrow argmin_V \; ||\Omega_v \mathbf{v} - b_v||^2 + \eta_v |\mathbf{v}|$
  $U \leftarrow argmin_U \; ||\Omega_u \mathbf{u} - b_u||^2 + \eta_u |\mathbf{u}|$ .

- End:
If $|\mathbf{v}^{(t+1)} - \mathbf{v}^{(t)}| + |\mathbf{u}^{(t+1)} - \mathbf{u}^{(t)}| < \varepsilon$ then stop.
Else return to E-Step.
```

Figure 2: Algorithm for learning the parameters in the binary blockpca with a Newton-Raphson or variational algorithm for the M-step.

Here, equating the gradient with zero leads to m equations. Matricially, this leads to a linear regression of the form $\Omega_v \mathbf{v} = b_v$ with the expression given at figure 1. The same expression is available for the variational approximation by replacing $b_\ell^{(t)}$ by $\tilde{b}_\ell^{(t)}$ and $H_\ell^{(t)}$ by $\tilde{H}_\ell^{(t)}$ while $\bar{v}_\ell^{(t)}$ depends on the algorithm. As in blockgtm, the linear regression may induce an eventual overfitting when the dimensions h, g or m increases. Recent developments in the literature have shown the good behavior of a parsimonious solution via a penalization of the criterion: the LASSO regression [18] adds a penalizing term $-\eta_v |\mathbf{v}|$ to the sum of squares in order to induce automatically the sparsity of the regression coefficients. At the convergence, all the quantities computed at the final step are denoted with a hat. The algorithm is summarized at the figure 2, by giving all the steps of the algorithm, from the initialization to the stopping function at the convergence with ε a small constant.

4 Conclusion

Herein it is explained how to model a pca method with a co-clustering for large sparse binary tables. The resulting reductions for the two dimensions of the table are written with the latent matrices $\hat{U}, \hat{V}$ and the posterior probabilities as follows:

$$
\begin{aligned}
\hat{x}_i &= \textstyle\sum_k \hat{c}_{ik} \hat{u}_k \\
\hat{x}^j &= \textstyle\sum_\ell \hat{d}_{j\ell} \hat{v}_\ell ,
\end{aligned}
$$

where $\hat{x}_i$ is the reduced i^{th} row and $\hat{x}^j$ is the reduced j^{th} column, as h-dimensional vectors where $h \ll n$ and $h \ll d$. A perspective is other types of variables [14] such as in contingency tables for textual data [12], more constraints such as orthogonality or alternative reducing parameterization.

References

[1] Benzecri, J.P.: Correspondence Analysis Handbook. New-York : Dekker (1992)

[2] Bock, H.: Simultaneous clustering of objects and variables. In: E. Diday (ed.) Analyse des Données et Informatique, pp. 187–203. INRIA (1979)

[3] Govaert, G.: Classification croisée. Thèse d'état, Université Paris 6, France (1983)

[4] Govaert, G.: Simultaneous clustering of rows and columns. Control and Cybernetics **24**(4), 437–458 (1995)

[5] Govaert, G., Nadif, M.: Clustering with block mixture models. Pattern Recognition **36**(2), 463–473 (2003)

[6] Govaert, G., Nadif, M.: An EM algorithm for the block mixture model. IEEE Trans. Pattern Anal. Mach. Intell. **27**(4), 643–647 (2005)

[7] Kohonen, T.: Self-organizing maps. Springer (1997)

[8] Lebart, L., Morineau, A., Warwick, K.: Multivariate Descriptive Statistical Analysis. J. Wiley (1984)

[9] Lee, S., Huang, J.Z.: A coordinate descent MM algorithm for fast computation of sparse logistic pca. Comput. Stat. Data Anal. **62**, 26–38 (2013)

[10] Lee, S., Huang, J.Z., Hu, J.: Sparse logistic principal components analysis for binary data. The Annals of Applied Statistics **4**(3), 1579–1601 (2010)

[11] de Leeuw, J.: Principal component analysis of binary data by iterated singular value decomposition. Comput. Stat. Data Anal. **50**(1), 21–39 (2006)

[12] Priam, R.: Negative binomial latent block model with generalized constraints (2021). Working paper or preprint

[13] Priam, R., Nadif, M., Govaert, G.: Topographic bernoulli block mixture mapping for binary tables. Pattern Analysis and Applications **17**(4), 839–847 (2014)

[14] Priam, R., Nadif, M., Govaert, G.: Generalized topographic block model. Neurocomputing **173**, 442–449 (2016)

[15] Saul, L.K., Jaakkola, T., Jordan, M.I.: Mean field theory for sigmoid belief networks. Journal of Artificial Intelligence Research **4**, 61–76 (1996)

[16] Schein, A.I., Lawrence, A.I., Saul, L.K., Ungar, L.H.: A Generalized Linear Model for Principal Component Analysis of Binary Data. In: AISTAT (2003)

[17] Symons, M.J.: Clustering criteria and multivariate normal mixture. Biometrics **37**, 35–43 (1981)

[18] Tibshirani, R.: Regression shrinkage and selection via the lasso. J. Royal. Statist. Soc B. **58**(1), 267–288 (1996)

[19] Tipping, M.E.: Probabilistic visualisation of high-dimensionnal binary data. Advances in Neural Information Processing Systems pp. 592–598 (1999)

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