STA561: Probabilistic machine learning

Factor Analysis (10/2/13)

Lecturer: Barbara Engelhardt *Scribes: Li Zhu, Fan Li, Ni Guan*

Factor Analysis

Factor analysis is related to the mixture models we have studied. One limitation of mixture models is that they only use a single latent variable to generate each observation; however, in real the world, data come from multiple clusters, for example:

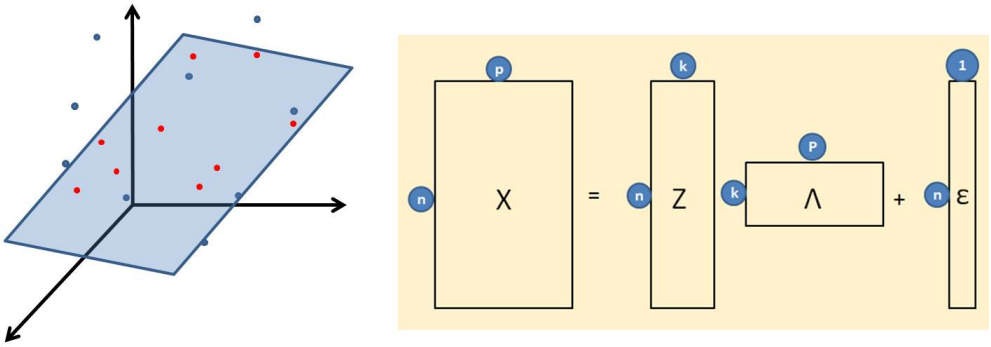
1. An Exam Question may test multiple topics, such as calculus, geometry, topology, and probability. Students' performance on each question is the matrix of observations (n students; p questions)
2. A picture include zero or multiple known objects: sunset, tree, mouse, cat, etc. Pixel features may be observed for each feature; (n images, p pixel features)
3. A document may be about multiple topics, including economy, government, education, sports, etc. The counts of different words in each document are the observed data; (n documents; one count for each word in the dictionary of p possible words)
4. Gene expression levels may be a function of a number of underlying variables (sample age, sample cell type, etc); (n samples, p genes).

Commonly, factor analysis is a method for *dimension reduction*: it reduces the dimension of the original space by representing the full matrix as the product of two much smaller matrices plus random noise. We are able to model the variability among the observed variables by projection from a high dimensional space to a low dimensional one, such from 3D to 2D as shown in Figure 1(a). To do this, let's start with a matrix of real-valued latent variables: $X \in \Re^{n \times p}$. The observed variables are modeled as a linear combination of the latent variables plus Gaussian error as follows:

$$X_{n \times p} = Z_{n \times k} \Lambda_{k \times p} + \epsilon_{n \times p}$$

Where,

1. $X_{n \times p} (x_i \in \mathbb{R}^p)$ is the observed data matrix, with n observations of p variables or *features*;
2. $Z_{n \times k} (z_i \in \mathbb{R}^k)$ is the factor matrix that includes information for the k factors for each of n observations;
3. $\Lambda_{k \times p}$ is the factor loading matrix that includes the weight of each feature (that has a loose interpretation, in the absolute value, as the contribution of that feature to that factor) for each factor;
4. ϵ is Gaussian noise, $\epsilon_i \sim N(0, \Psi)$. Note that the noise is in the p space, not the lower dimensional space.



(a) (b)

Figure 1: Graphical representation of Factor Analysis (a). A three-dimensional example, projecting down to a Gaussian ball around a two-dimensional hyperplane (a linear subspace); (b). Matrix representation of Factor Analysis (actually, the figure is incorrect; the noise is $n \times p$, not a vector).

Factor analysis is an *exploratory data analysis* method that can be used to discover a small set of components that underlie a high-dimensional data set. It has many purposes:

- Dimension reduction: reduce the dimension of (and denoise) a high-dimensional matrix
- Linear projection: project this high dimensional matrix down to a low dimensional linear subspace
- Matrix factorization: factorize a high dimensional matrix into two low dimensional matrices, where orthogonality between the factors is not required

This has parallels with linear regression except that there is no vector of observed predictors; instead there is only a response matrix that has been observed. As in linear regression, we can include a mean term (parallel to the intercept term in the β coefficients in linear regression) that corresponds to the means of each of the p features. This is equivalent to adding a column of 1s in the matrix Λ, and a corresponding row to matrix Z that has the interpretation of the mean terms. In deriving the EM updates for the parameters, though, it is easier to consider this vector of mean parameters, μ, separately from the low dimensional matrices.

Marginal likelihood of a single sample

Factor analysis is a probabilistic model of the joint density of matrix X using a small(er) number of parameters. Relying on this probabilistic framework, we can determine the marginal likelihood of a single sample X_i as follows. We will put a Gaussian prior on the real-valued latent factors ($P(z_i) \sim N(z_i|\mu_0, \Sigma_0)$), and integrate these factors out to obtain the marginal distribution of X_i.

$$\begin{aligned} P(x_i|\Lambda, \Psi, \mu_0) &= \int P(x_i|\Lambda, z_i, \mu, \Psi) P(z_i|\mu_0, \Sigma_0) dz_i \\ &= N_p(x_i|\Lambda\mu_0 + \mu_i, \Psi + \Lambda^T \Sigma_0 \Lambda) \end{aligned}$$

If $\Sigma_0 = I$, the the previous equation could be written as:

$P(x_i|\Lambda, \Psi, N) = N_p(x_i|\mu_i, \Psi + \Lambda^T\Lambda)$.

Note that the marginal distribution of X_i is Gaussian. To simplify this further, we could set $\mu_0 = 0$ without loss of generality, since $\Lambda\mu_0$ will be absorbed into μ. Similarly, Σ_0 could be set to the identity matrix I without loss of generality, because we can always "emulate" a correlated prior by using defining a new weight matrix $\tilde{\Lambda} = \Lambda\Sigma_0^{-\frac{1}{2}}$, then:

$$\begin{aligned} cov[x|\theta] &= \tilde{\Lambda}\tilde{\Lambda}^T + \mathbb{E}[\epsilon\epsilon^T] \\ &= (\Lambda\Sigma_0^{-\frac{1}{2}})\Sigma_0(\Lambda\Sigma_0^{-\frac{1}{2}})^T + \Psi \\ &= \Lambda\Lambda^T + \Psi \end{aligned}$$

in which:

1. $\Lambda\Lambda^T$ models the covariance structure of the original matrix X in dimension p

2. Ψ models the variance structure. Ψ is not required to be diagonal, but when it is, Λ is used to model the covariance of the original matrix.

Because of these features of the FA model, it is often thought of as a low dimensional representation of the covariance matrix of X.

Example: Interpretation of the Exam Question example

How can we explore this low dimensional representation of our high dimensional matrix X? Let us look at an explicit example. As above, let X represent n students' answers for p exam questions. We will estimate the matrices Λ and Z for this matrix with K factors.

In Figure 2, X carries the n students' performance on each of p questions on the exam. Figure 2 highlights how the factors and the factor loading matrix explain x_{ij} (the i^{th} student's response to the j^{th} question, where $i = 1, ..., n$ and $j = 1, ..., p$). The i^{th} student's performance is recorded as x_i and can be interpreted through z_i and Λ.

- z_i is relative performance of a student for each underlying topic; in this example z_i can represent how well a student answers questions about geometry, calculus, topology, or word problems (if those are the interpretations of the underlying topics)

- Λ is the relative influence of each question on the underlying topics; in this example Λ can be thought of the correlation of each of the p questions to each of the k topics in z_i.

For example in part b), z_i might show that the student is very bad at topic number 5 and very good at topic number 2 (keep in mind that these matrices are *sign invariant*, so we must look at the data to determine which topic this student is good and bad at; the idea is that any factor that has a large magnitude the student will deviate from the global average performance on questions pertaining to that topic). In part c), Λ_j reflects

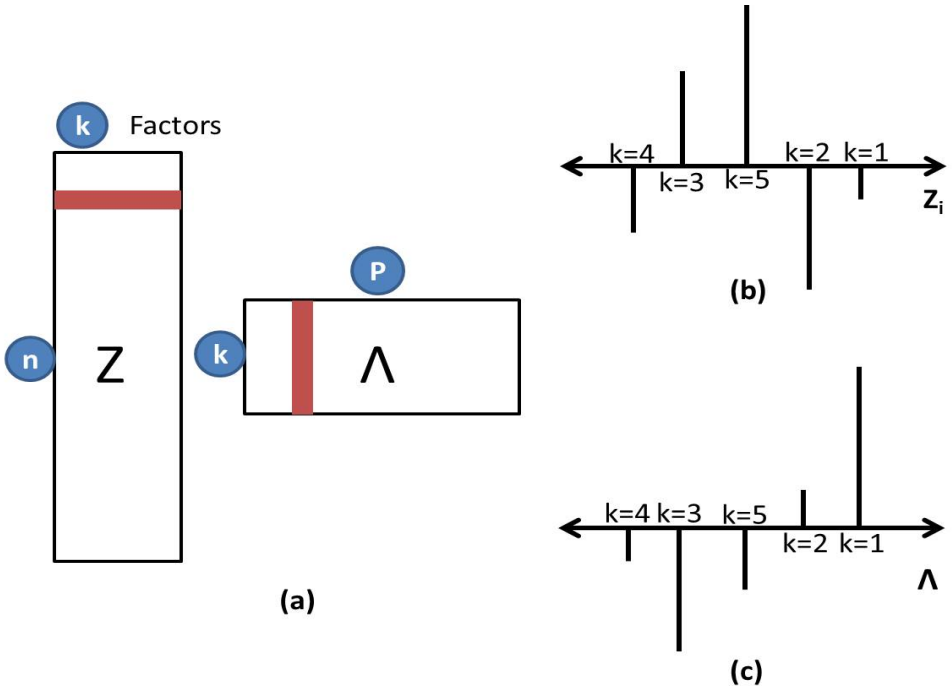


Figure 2: Visual interpretation of 'Exam Question' example. The plots in parts b) and c) are vectors indicated by the red lines from the factor matrix and factor loading matrix in part a) respectively.

how each underlying topics contribute to the j^{th} question. In this case the 1^{st} and 3^{rd} topics contribute to the j^{th} question, while the 4^{th} topic only has minimal contribution.

As with mixture models, the interpretations or labels of the underlying topics are user-defined. In other words, to determine that a specific topic captures questions related to *geometry*, we might look at the relative contribution of all of the exam questions (via the estimated Λ matrix) and notice that those with the greatest contribution were generally about geometry, and those with the least contribution had no geometry. But this is a manual process (as we've defined it), and these labels represent an additional layer of interpretation on top of this exploratory analysis.

Unidentifiability

Unidentifiability is a problem in factor analysis, in several different ways. Here we present several situations where we may not uniquely identify the factor loading matrix and the latent factors, as well as possible solutions to deal with them.

Unidentifiability A: Unidentifiability up to orthogonal rotation

Suppose that $\mathbf{R}$ is an arbitrary $k \times k$ orthogonal rotation matrix satisfying $\mathbf{R}\mathbf{R}^T = \mathbf{I}$, then define $\tilde{\mathbf{\Lambda}} = \mathbf{R}\mathbf{\Lambda}$, and also rotate the latent factors as:

$$\tilde{\mathbf{Z}} = \mathbf{Z}\mathbf{R}^T.$$

Then we have:

$$\tilde{\mathbf{Z}}\tilde{\mathbf{\Lambda}} = \mathbf{Z}\mathbf{R}^T\mathbf{R}\mathbf{\Lambda} = \mathbf{Z}\mathbf{\Lambda}.$$

Hence the mean of data x_i conditioned on latent factors z_i does not change under rotation, and we fail to uniquely identify the factor loadings and the corresponding latent factors up to orthogonal rotation.

Also note that the covariance matrix for $\mathbf{x}$ is:

$$\text{cov}[\mathbf{x}] = \Psi + \tilde{\mathbf{\Lambda}}^T\tilde{\mathbf{\Lambda}} = \Psi + \mathbf{\Lambda}^T\mathbf{R}^T\mathbf{R}\mathbf{\Lambda} = \Psi + \mathbf{\Lambda}^T\mathbf{\Lambda}$$

After rotation, the covariance matrix of $\mathbf{x}$ does not change, the marginally data likelihood is invariant to this rotation of factor loadings. Again we are not able to identify $\mathbf{\Lambda}$ up to orthogonal rotation.

Solutions might be:

- *Force $\mathbf{\Lambda}$ to be orthonormal.* This might be the cleanest solution to identifiability issues since by forcing the orthonormal property and ordering the columns of $\mathbf{\Lambda}$ in order of decreasing variance of the corresponding latent factors, we are putting more constraints on $\mathbf{\Lambda}$ hence identifiability becomes possible. This is the approach that Principal Components Analysis adopts. The result may not be easily interpretable, but at least $\mathbf{\Lambda}$ and $\mathbf{Z}$ are unique.

- *Force $\mathbf{\Lambda}$ to be lower triangular.* This way to achieve identifiability is popular in Bayesian community. The intuition is to achieve identifiability by ensuring that the first observed feature is only generated from the first latent factor, and the second observed feature is only generated through the first two latent factors and so on. Here we briefly present the mechanism and why it works in reality. Recall from matrix algebra, the orthogonal rotation matrix $\mathbf{R}_{k\times k}$ has $1+2+...+(k-1) = \frac{k(k-1)}{2}$ free parameters, and kp parameters are used to characterize $\mathbf{\Lambda}$. The identifiability issues come from orthogonal rotation, so we need to subtract $\frac{k(k-1)}{2}$ from kp and use this many parameters to uniquely identify the matrix. This can be thought of as the effective degree of freedom in statistics to achieve model identifiability. If we force $\mathbf{\Lambda}$ to be lower triangular, we actually have $\frac{k(k-1)}{2}$ zero inputs in the upper triangular part of the matrix, and we have the right amount of parameters $kp - \frac{k(k-1)}{2}$ to uniquely characterize $\mathbf{\Lambda}$. For example, if $k = 4$ and $p = 3$, we might set

$$\mathbf{\Lambda} = \begin{pmatrix} \lambda_{11} & 0 & 0 \\ \lambda_{21} & \lambda_{22} & 0 \\ \lambda_{31} & \lambda_{32} & \lambda_{33} \\ \lambda_{41} & \lambda_{42} & \lambda_{43} \end{pmatrix}$$

- *Use sparsity promoting priors on $\mathbf{\Lambda}$.* The previous approach pre-defines which entries of $\mathbf{\Lambda}$ should be zero, but we could actually use sparsity encouraging priors on $\mathbf{\Lambda}$ through methods such as l_1 regularization, automatic relevance determination or a spike-and-slab prior. More on this in the next class.

- *Choose an informative rotation matrix.* By choosing an informative rotation matrix $\mathbf{R}$, the interpretation could be easier. One popular method is known as **varimax** (Kaiser 1958), where we choose the rotation that includes the maximal number of matrix loading elements near zero.

- *Use non-Gaussian priors for the latent factors.* Choosing non-Gaussian priors on $\mathbf{z_i}$ could help us uniquely identify $\mathbf{\Lambda}$ sometimes as well as $\mathbf{Z}$, this technique is usually called Independent Component Analysis (ICA).

Unidentifiability B: Unidentifiability up to scaling

Apart from orthogonal rotation, we could also simultaneously scale the factor loadings as well as the latent factors by a constant α. With this, we still have the issue of unidentifiability:

$$\mathbf{z}_k \mathbf{\Lambda}_k = (\mathbf{z}_k \frac{1}{\alpha})(\mathbf{\Lambda}_k \alpha)$$

Solutions might be:

- *Force $\mathbf{\Lambda}$ to be orthonormal (or just normalize $\mathbf{\Lambda}$).* As in Unidentifiability A, this is a clean solution since we are adding constraints to find a unique $\mathbf{\Lambda}$ and $\mathbf{Z}$.
- *Avoid consideration of the loadings or factors in side-by-side comparison.* Think instead of within-factor magnitude.

Unidentifiability C: Unidentifiability up to label switching

Suppose we have K latent factor variables. Let's say that we have two different orderings of the rows of $\mathbf{\Lambda}$ and their corresponding columns of $\mathbf{z}$ but identical product:

$$\mathbf{z}_k \mathbf{\Lambda}_k = \mathbf{z}_{k'} \mathbf{\Lambda}_{k'}$$

$$k \in \{2, 3, 1\} \qquad k' \in \{1, 2, 3\}$$

This is known as the label-switching problem, since cannot distinguish between the two orderings based on likelihood; moreover, we can see that the factor loading matrix is not comparable across rows since ordering affects the unique identification of the factor loadings.

Solutions might be:

- *Put a prior on the Percentage of Variance Explained by each factor.* Methods (such as PCA) will produce a more robust ordering of the latent factors.
- *Avoid matching factors/loadings across runs.* Compare instead based, for example, on covariance matrix estimates.

EM for Factor Analysis

Similar to what we have done with the mixture models in the previous lectures, we can derive the EM algorithm for the factor analysis model. For simplicity, we assume our model and observed data have zero mean ($\mu = 0$, in the midterm you derive when $\mu \neq 0$).

$$\begin{aligned} z &\sim \mathcal{N}(0, I) \\ \epsilon &\sim \mathcal{N}(0, \Psi) \\ x &= \Lambda z + \epsilon \\ x|z &\sim \mathcal{N}(\Lambda z, \Psi) \end{aligned}$$

where x is a $p \times 1$ vector containing our data , z is a $k \times 1$ vector containing our factors, and ϵ the error term. Then, variables X and Z are jointly normal, and, from the model definition, we can derive the joint distribution as

$$\begin{bmatrix} x \\ z \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \Lambda^T\Lambda + \Psi & \Lambda^T \\ \Lambda & I \end{bmatrix} \right).$$

By the definition of the multivariate normal distribution, we have the conditional expectation and variance terms:

$$\mathbb{E}[z|x] = \Lambda^T(\Lambda\Lambda^T + \Psi)^{-1}x$$
$$\mathbb{V}[z|x] = I - \Lambda^T(\Lambda\Lambda^T + \Psi)^{-1}\Lambda$$

In the **E-step**, we would like to take the expectation of the Z variables. The expected log likelihood we want to maximize is

$$\mathbb{E}\sum_{i=1}^{n} \log P(x_i|z_i, \Lambda, \Psi) = \sum_{i=1}^{n} \mathbb{E}[\log \frac{1}{(2\pi)^{p/2}|\Psi|^{1/2}} \exp(-\frac{1}{2}(x_i - \Lambda z_i)^T\Psi^{-1}(x_i - \Lambda z_i))]$$

Dropping terms that do not depend on the parameters, we need to maximize:

$$Q(z_i) = -\frac{n}{2}\log|\Psi| - \sum_{i=1}^{n}(\frac{1}{2}x_i^T\Psi^{-1}x_i - x_i^T\Psi^{-1}\Lambda^T\mathbb{E}(z|x_i) + \frac{1}{2}trace[\Lambda\Psi^{-1}\Lambda^T\mathbb{E}[z^Tz|x_i]]).$$

We have solved $\mathbb{E}[z|x]$ and $\mathbb{V}[z|x]$ previously, and $\mathbb{E}[z^Tz|x]$ can be derived as

$$\begin{aligned} &\mathbb{E}[zz^T|x] = \mathbb{E}[z|x]\mathbb{E}[z|x]^T + \mathbb{V}[z|x] \\ &= \Lambda^T(\Lambda\Lambda^T + \Psi)^{-1}xx^T(\Lambda^T\Lambda + \Psi^T)^{-1}\Lambda + I - \Lambda^T(\Lambda\Lambda^T + \Psi)^{-1}\Lambda \end{aligned}$$

These equations will allow us to compute the expected sufficient statistics for our FA model.

For the **M-step**, first take the derivative of $Q(z_i)$ with respect to parameter Λ and set it to zero, and solve for Λ:

$$\frac{\partial Q}{\partial \Lambda} = -\sum_{i=1}^{n}\Psi^{-1}x_i\mathbb{E}[z_i|x_i]^T + \sum_{i=1}^{n}\Psi^{-1}\Lambda\mathbb{E}[zz^T|x_i],$$

obtaining

$$\hat{\Lambda} = (\sum_{i=1}^{n}x_i\mathbb{E}[z|x_i]^T)(\sum_{i=1}^{n}\mathbb{E}[zz^T|x_i])^{-1}.$$

Then, to derive the updates to Ψ, we take the derivative of $Q(z_i)$ with respect to the inverse of Ψ and set it to zero and solve:

$$\frac{\partial Q}{\partial \Psi^{-1}} = \frac{n}{2}\Psi - \sum_{i=1}^{n}(\frac{1}{2}x_ix_i^T - \Lambda\mathbb{E}[z|x_i]x_i^T + \frac{1}{2}\Lambda\mathbb{E}[zz^T|x_i]\Lambda^T) = 0$$

obtaining

$$\hat{\Psi} = \frac{1}{n} diag\{\sum_{i=1}^{n} x_i x_i^T - \Lambda \mathbb{E}[z|x_i] x_i^T\}$$

where $diag\{\}$ operator only choose the diagonal elements of that matrix. The full derivation of the EM algorithm for factor analysis can be found in, e.g., (Ghahramani and Hinton 1996).

Additional Material (see course website for a list)

A Unifying Review of Linear Gaussian Models (Sam Roweis and Zoubin Ghahramani, 1999). In this comprehensive review paper, the authors unified many models we have learned as variants of unsupervised learning using a generative model and proposed a new model SPAC (sensible principal components analysis) for static data.

MacKay: Chapter 34 (Independent Component Analysis and Latent Variable Modeling). Provides detailed account of ICA and relevant latent variable models.

Metacademy Series: Factor Analysis and Principal Components Analysis. Alternative materials on FA and PCA, free resources about popular book materials *Bayesian Reasoning and Machine Learning* on FA, and Coursera materials on PCA by Andrew Y. Ng with free text *The Elements of Statistical Learning*.