Matrix Factorization

David M. Blei
Columbia University

December 4, 2016

1 Dyadic data

One important type of modern data is *dyadic data*. Dyadic data are measurements on pairs. The idea is that the observed value y_{ij} represents something measured about the interaction of i and j.

One of the most prevalent types of dyadic data is *user consumption data*, where y_{ij} represents an interaction between user i and item j. For example, y_{ij} can be the number of times user i clicked on webpage j, whether user i purchased book j, the rating user i gave to movie j, or how long user i spent at location j. This problem has cropped up everywhere: Amazon (purchasing), web sites (clicks), music listening services (listens). It is important both for forming predictions & for understanding people's behavior. Throughout this lecture, we will use the language of user consumption behavior: there are users, items, and measurements about them.

Typically these data are represented in a matrix, the *consumption matrix*.

¶ [DRAW A MATRIX, MOVIES AND USERS; STAR WARS AND ROMCOMS]

As usual, the two types of problems we want to solve with dyadic data are prediction and exploration. In exploration, we want to understand patterns of consumption—what kind of user likes what kind of movie? These problems are less explored in Computer Science, but they are very important in Econometrics.

Here we focus on prediction. In prediction we want to predict which unrated items a user will like. For example, will user #3 like Star Wars? Empire Strikes Back? When Harry Met Sally? (How about the new Star Wars movie? Already this gets interesting. That prediction problem is called the *cold start problem*. More on that

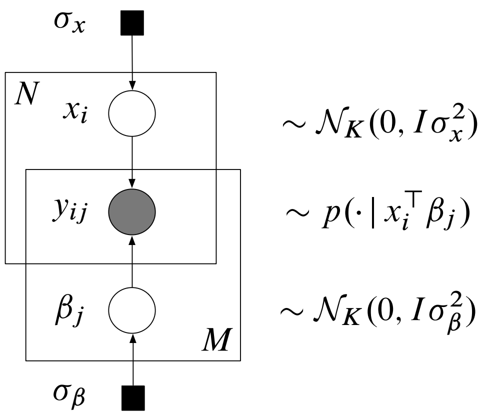
σ_x
N
x_i
$\sim \mathcal{N}_K(0, I\sigma_x^2)$
y_{ij}
$\sim p(\cdot \mid x_i^\top \beta_j)$
β_j
$\sim \mathcal{N}_K(0, I\sigma_\beta^2)$
M
σ_β

Figure 1: The graphical model for probabilistic matrix factorization.

later.) More realistically, in prediction we provide *ranked lists* of unwatched movies for the users. A good list has movies they will like ranked higher in the list.

2 Matrix Factorization

In probabilistic matrix factorization, each user and each item is associated with a latent vector. For users we call this vector her *preferences*; for items we call this vector its *attributes*. Loosely, whether a user likes an item is governed by the distance between a user's preferences and the item's attributes. If they are closer together in the latent space, then it is more likely that the user will like the item.

¶ [DRAW PICTURE IN 2D; INCLUDE STAR WARS AND ROMCOMS]

Formally, the generative process is

1. For each user i, draw each preference $x_{ik} \sim \mathcal{N}(0, \sigma_x^2)$, where $k \in 1, \ldots, K$.
2. For each item j, draw each item attribute $\beta_{ik} \sim \mathcal{N}(0, \sigma_\beta^2)$, where $k \in 1, \ldots K$.
3. For each user item pair, draw $y_{ij} \sim f(x_i^\top \beta_j)$.

Figure 1 gives the graphical model. (For those that are familiar, this is similar to factor analysis and principal component analysis.)

Given a user/item matrix, our goal is to compute the posterior distribution of x_i (for each user) and β_j (for each item). This embeds users and items into a low-dimensional space. It places users close to the items that they like; and places items close to the users who like them.

Intuitively, why does this work? Consider two users, Jack and Diane. Jack has seen every Star Wars movie except for Return of the Jedi. Diane has seen every Star Wars movie. The posterior will want to place Jack near the movies he saw and place Diane near the movies that she saw. But they have seen similar collections of movies. Thus they will be near each other in the latent space, and near the Star Wars movies. Consequently, Return of the Jedi will be close to Jack, and we can recommend that he see it.

As we did for LDA, let's write down the log of the joint distribution. This will help us make these intuitions more rigorous.

The joint distribution of hidden and observed variables is

$$\log p(\boldsymbol{\beta}, \mathbf{x}, \mathbf{y}) = \sum \log p(x_i) + \sum \log p(\beta_j) + \sum \sum \log p(y_{ij} \mid x_i, \beta_j). \quad (1)$$

Assume the data are Gaussian,

$$y_{ij} \mid x_i, \beta_j \sim \mathcal{N}(x_i^\top \beta_j, \sigma^2). \quad (2)$$

Thus the last term of the log joint is related to the mean squared error between observed ratings y_{ij} and the model's representation $x_i^\top \beta_j$.

Again, consider a set of users who like the same two movies. The model can place the movies close together in the latent space, place the users in the same vicinity, and explain the ratings. When new users like one of those movies, they are placed in the same part of the space; we predict that they like the other movie too.

The priors in the first two two terms in the log joint are *regularization terms*; they shrink the representations of users and movies, pulling them towards zero. Empirically, regularization is crucial in these models. Practitioners use cross-validation to set the regularization parameters, i.e., the hyperparameters to the prior. When we study regression, we will learn more about regularization and why it is important.

We turn to inference. First, we discuss the MAP estimates of preferences **x** and attributes ¿. (Recall the MAP estimate is the value of the latent variable that maximizes the posterior.)

We expand the log joint to include the Gaussian assumptions and remove terms that do not depend on the latent variables,

$$\mathcal{L} = \sum_k \left(\sum_i -x_{ik}^2/2\sigma_x^2 + \sum_j -\beta_{ik}^2/2\sigma_\beta^2 \right) + \sum_j -(y_{ij} - x_i^\top \beta_j)^2/2. \tag{3}$$

(To keep it simpler, we assumed that the likelihood variance is one.)

We take the derivative with respect to x_{ik}, user i's preference for factor k. We expand the square in the likelihood term and collect terms that depend on x_{ik},

$$\mathcal{L}(x_{ik}) = -x_{ik}^2/2\sigma_x^2 + \sum_j x_{ik}\beta_{jk}y_{ij} - (x_i^\top \beta_j)^2/2. \tag{4}$$

The derivative with respect to x_{ik} is

$$\begin{aligned} \tfrac{\partial \mathcal{L}}{\partial x_{ik}} &= -x_{ik}/\sigma_x^2 + \sum_j y_{ij}\beta_{jk} - (x_i^\top \beta_j)\beta_{jk} & (5) \\ &= -x_{ik}/\sigma_x^2 + \sum_j (y_{ij} - x_i^\top \beta_j)\beta_{jk} & (6) \\ &= -x_{ik}/\sigma_x^2 + \sum_j (y_{ij} - \mathbb{E}\left[Y_{ij} \,|\, x_i, \beta_j\right])\beta_{jk} & (7) \end{aligned}$$

For those of you that know regression, this is a (regularized) linear regression, one per user. The coefficients are x_i; the covariates are β_j; the response is y_{ij}. (There is one response per item.) We will see these types of gradients again when we study generalized linear models.

The derivative with respect to β_{jk} is analagous:

$$\tfrac{\partial \mathcal{L}}{\partial \beta_{jk}} = -\beta_{jk}/\sigma_\beta^2 + \textstyle\sum_i (y_{ij} - \mathbb{E}\left[Y_{ij} \,|\, x_i, \beta_j\right])x_{ik}. \tag{8}$$

This is a (regularized) regression—one per *item*—where β_j are the coefficients, x_i are the covariates, and y_{ij} is the response. (There is one response per user.)

Coordinate updates of these latent variables, iterates between solving user-based reguarlized regressions and item-based regularized regressions.

Now consider MCMC. The complete conditional of the preference x_{ik} is proportional to the joint. Examining Equation 4, we see that this is a Gaussian distribution. It is an exponential family with sufficient statistics x_{ik} and x_{ik}^2.

We have scratched the surface. Further ideas:

- Tensor factorization
- The latent space model of networks
- Including observed characteristics
- Big challenges: sparsity, implicit data
- The cold start problem and content-based recommendation