Exercises 9: Matrix factorization

Before starting, make sure to read the paper "A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis," by Witten, Tibshirani, and Hastie, linked through the class website.

A sparse matrix factorization

Suppose that X is a large $N \times P$ data matrix. Each row of X corresponds to multiple features/measurements about a single subject (e.g. person, gene, household, etc). The goal is to describe a small set of latent factors that describe a lot of the correlation structure among the features. As in principal components analysis, we'll approach this problem from the standpoint of matrix factorization.

Consider the problem of finding a rank-K approximation to a data matrix X of the form

$$\hat{X} = \sum_{k=1}^{K} d_k u_k v_k^T \, ,$$

where $d_k > 0$ is a scalar, and $u_k \in \mathcal{R}^N$ and $v_k \in \mathcal{R}^P$ are vectors (sometimes called "left factors" and "right factors," respectively. If we stack these vectors in matrices, we can write

$$\hat{X} = UDV^T \, ,$$

where U has the u_k vectors along each column, V has the v_k vectors along each column, and $D = \mathrm{diag}(d_1, \ldots, d_K)$.

Ordinary PCA takes the u and v vectors to be the left- and right-singular vectors of the data matrix, respectively. But what if we want the u and v vectors to be sparse—that is, we want each one to involve only a small number of features of the original X matrix?

Let's first consider the case of a single factor, $K = 1$, so that $\hat{X} = duv^T$. This gives us a rank-1 approximation to the original matrix. The Witten et. al. paper proposes to estimate u and v by solving the following optimization problem:

$$\begin{aligned} \underset{u \in \mathbb{R}^N, v \in \mathcal{R}^P}{\text{minimize}} \quad & \|X - duv^T\|_F^2 \\ \text{subject to} \quad & \|u\|_2^2 = 1 \, , \; \|v\|_2^2 = 1 \, , \\ & \|u\|_1 \leq \lambda_u \, , \; \|v\|_1 \leq \lambda_v \, , \end{aligned} \tag{1}$$

where $\| \cdot \|_F^2$ is the squared Frobenius norm of a matrix (that is, the sum of squared elements of the matrix). Note that if we remove the

ℓ_1 constraints on u and v, the solution to the problem is exactly the first left- and right-singular vectors of the data matrix. In other words, we just recover the first principal component. However, solving the problem with the ℓ_1 constraints imposed will lead to a sparse (and presumably more interpretable) u and v.

Write your own code, following the exposition of Witten et. al., that is capable of solving this sparse rank-1 factorization problem for a given choice of λ_u and λ_v. Simulate some data to make sure that your implementation is behaving sensibly. Note: to estimate factors 2, 3, and so on, we recursively apply the rank-1 factorization to the residual matrix from the previous stage.

Application to marketing

Consider the data in "social-marketing.csv." This was data collected in the course of a market-research study using followers of the Twitter account of a large consumer brand that shall remain nameless—let's call it NutrientH20 just to have a label. The goal here was for NutrientH20 to understand its social-media audience a little bit better, so that it could hone its messaging a little more sharply.

A bit of background on the data collection: the advertising firm who runs NutrientH20's online-advertising campaigns took a sample of the brand's Twitter followers. They collected every Twitter post (tweet) by each of those followers over a seven-day period in June 2014. Every post was examined by a human annotator contracted through Amazon's Mechanical Turk service. Each tweet was categorized based on its content using a pre-specified scheme of 36 different categories, each representing a broad area of interest (e.g. politics, sports, family, etc.) Annotators were allowed to classify a post as belonging to more than one category. For example, a post such as "I'm really excited to see grandpa go destroy the opposition in his geriatic soccer league this Sunday!" might be categorized as both family and sports. You get the picture.

Each row of "social-marketing.csv" represents one user, labeled by a random (anonymous, unique) 9-digit alphanumeric code. Each column represents an interest, which are labeled along the top of the data file. The entries are the number of posts by a given user that fell into the given category. Two interests of note here are "spam" (i.e. unsolicited advertising) and "adult" (posts that are pornographic or explicitly sexual). There are a lot of spam and pornography "bots" on Twitter; while these have been filtered out of the data set to some extent, there will certainly be some that slip through. There's also an "uncategorized" label.

Annotators were told to use this sparingly, but it's there to capture posts that don't fit at all into any of the listed interest categories. (A lot of annotators may used the chatter category for this as well.) Keep in mind as you examine the data that you cannot expect perfect annotations of all posts. Some annotators might have simply been asleep at the wheel some, or even all, of the time! Thus there is some inevitable error and noisiness in the annotation process.

Use the matrix factorization you have just implemented to find any interesting market segments that appear to stand out in this social-media audience. Note: you might want to apply the factorization to a transformed version of the original data matrix. I leave it to you to decide what kind of transformation or scaling is appropriate here.