6. Mixture models for matrix completion

We can now extend our Gaussian mixture model to predict actual movie ratings. Let X again denote the (n, d) data matrix. The rows of this matrix correspond to users and columns specify movies so that `X[u,i]` gives the rating value of user u for movie i (if available). Both n and d are typically quite large. The ratings range from one to five stars and are mapped to integers $\{1, 2, 3, 4, 5\}$. We will set `X[u,i]=0` whenever the entry is missing.

In a realistic setting, most of the entries of X are missing. For this reason, we define C_u as the set of movies (column indexes) that user u has rated and H_u as its complement (the set of remaining unwatched/unrated movies we wish to predict ratings for). We use $|C_u|$ to denote the number of observed rating values from user u. From the point of view of our mixture model, each user u is an example $x^{(u)} =$ `X[u,:]`. But since most of the coordinates of $x^{(u)}$ are missing, we need to focus the model during training on just the observed portion. To this end, we use $x_{C_u}^{(u)} = \{x_i^{(u)} : i \in C_u\}$ as the vector of only observed ratings. If columns are indexed as $\{0, \ldots, d-1\}$, then a user u with a rating vector $x^{(u)} = (5, 4, 0, 0, 2)$, where zeros indicate missing values, has $C_u = \{0, 1, 4\}$, $H_u = \{2, 3\}$, and $x_{C_u}^{(u)} = (5, 4, 2)$.

In this part, we will extend our mixture model in two key ways.

- First, we are going to estimate a mixture model based on partially observed ratings. See notes below.
- Second, since we will be dealing with a large, high-dimensional data set, we will need to be more mindful of numerical underflow issues. To this end, you should perform most of your computations in the log domain. Remember, $\log(a \cdot b) = \log(a) + \log(b)$. This can be useful to remember when a and b are very small – in these cases, addition should result in fewer numerical underflow issues than multiplication.

  An additional numerical optimization trick that you will find useful is the LogSumExp trick. Assume that we wish to evaluate $y = \log(\exp(x_1) + \ldots \exp(x_n))$ We define $x^* = \max\{x_1 \ldots x_n\}$. Then, $y = x^* + \log(\exp(x_1 - x^*) + \ldots \exp(x_n - x^*))$. This is just another trick to help ensure numerical stability.

Marginalizing over unobserved coordinates

If $x^{(u)}$ were a complete rating vector, the mixture model from Part 1 would simply say that $P(x^{(u)}|\theta) = \sum_{j=1}^{K} \pi_j N(x^{(u)}; \mu^{(j)}, \sigma_j^2 I)$. In the presence of missing values, we must use the marginal probability $P(x_{C_u}^{(u)}|\theta)$ that is over only the observed values. This marginal corresponds to integrating the mixture density $P(x^{(u)}|\theta)$ over all the unobserved coordinate values. In our case, this marginal can be computed as follows.

The mixture model for a complete rating vector is written as:

$$P(x^{(u)}|\theta) = \sum_{j=1}^{K} p_j N(x^{(u)}; \mu^{(j)}, \sigma_j^2 I)$$

We can decompose the multivariate spherical Gaussian as a product of univariate Gaussians (since there is no covariance between coordinates).

$$\begin{aligned} P(x^{(u)}|\theta) &= \sum_{j=1}^{K} p_j \prod_i N(x_i^{(u)}; \mu_i^{(j)}, \sigma_i^{2,(j)}) \\ &= \sum_{j=1}^{K} p_j \prod_{m \in C_u} N(x_m^{(u)}; \mu_m^{(j)}, \sigma_m^{2,(j)}) \prod_{m' \in H_u} N(x_{m'}^{(u)}; \mu_{m'}^{(j)}, \sigma_{m'}^{2,(j)}) \end{aligned}$$

For $m' \in H_u$, we can marginalize over all of the unobserved values to get

$$\int N\left(x_{m'}^{(u)}; \mu_{m'}^{(j)}, \sigma_{m'}^{2,(j)}\right) dx_{m'}^{(u)} = 1$$

Thus, our mixture density can be written as

$$P\left(x_{C_u}^{(u)}|\theta\right) = \sum_{j=1}^{K} p_j N\left(x_{C_u}^{(u)}; \mu_{C_u}^{(j)}, \sigma_j^2 I_{C_u \times C_u}\right)$$

Discussion

Show Discussion

Topic: Unit 4 Unsupervised Learning (2 weeks) :Project 4: Collaborative Filtering via Gaussian Mixtures / 6. Mixture models for matrix completion