Factor Model for Categorical Data

Lawrence Carin
Electrical and Computer Engineering Department
Duke University
Durham, NC 27708
lcarin@duke.edu

A. Model

Consider a setting for which we have N questions, and for question $n \in \{1, \ldots, N\}$ we have $K_n \geq 2$ possible categorical answers. The observed answer for article m and question n is $a_{m,n} \in \{1, \ldots, K_n\}$. Assume we have M news articles, and for each article m we have the set of answers $\{a_{m,n}\}_{n=1,N}$.

Within our model, for each question n with K_n possible categorical answers, we have a set of $K_n - 1$ vectors $\{w_{n,k}\}_{k=1,K_n-1}$, where each $w_{n,k} \in \mathbb{R}^d$, where $d > 1$ is a chosen latent dimension. Further, assume that article m is characterized by a latent vector $v_m \in \mathbb{R}^d$. Then we model the probability of answer $a_{m,n} = k$ as

$$p_\theta(a_{m,n} = k|v_m) = \frac{\exp[w_{n,k}^T v_m + b_{n,k}]}{1 + \sum_{k'=1}^{K_n-1} \exp[w_{n,k'}^T v_m + b_{n,k'}]}, \quad \forall \quad k \in \{1, \ldots, K_n - 1\} \tag{1}$$

$$p_\theta(a_{m,n} = K_n|v_m) = \frac{1}{1 + \sum_{k'=1}^{K_n-1} \exp[w_{n,k'}^T v_m + b_{n,k'}]} \tag{2}$$

The model parameters for the N questions are $\theta = \{\{(w_{n,k}, b_{n,k})\}_{k=1,K_n-1}\}_{n=1,N}$. The model parameters θ are shared (are the same) across all M articles. For each article m the underlying latent vector $v_m \in \mathbb{R}^d$ that gives rise to the observed answers.

Based on M articles, and the N answers for each, our goal is to learn θ and $\{v_m\}_{m=1,M}$, and the closeness of articles (matching) will be performed by calculating the Euclidian distances between different v_m. The feature vector v_m is meant to characterize the topical content of article m.

B. Learning

Assume that we are given data corresponding to answers $\mathcal{D} = \{\{a_{m,n}\}_{n=1,N}\}_{m=1,M}$ to N categorical questions for M articles. We wish to infer θ and $\{v_m\}_{m=1,M}$. We define the following loss, associated with the negative log-likelihood of our model (or cross entropy):

$$\mathcal{L}(\theta, \{v_m\}_{m=1,M}) = -\sum_{m=1}^{M} \sum_{n=1}^{N} \log p_\theta(a_{m,n}|v_m) \tag{3}$$

and we perform the optimization-based estimation

$$\hat{\theta}, \{\hat{v}_m\}_{m=1,M} = \text{argmin}_{\theta, \{v_m\}_{m=1,M}} \left[\mathcal{L}(\theta, \{v_m\}_{m=1,M}) + \lambda_1 \sum_{n=1}^{N} \sum_{k=1}^{K_n-1} \|w_{n,k}\|_2 + \lambda_2 \sum_{m=1}^{M} \|v_m\|_2\right] \tag{4}$$

where $\lambda_1 > 0$ and $\lambda_2 > 0$ are scalar regularization parameters that are to be set. The ℓ_2 regularization on the model parameters is meant to aid model identifiability, and other such regularization may be considered (for example to infer the proper latent dimension d); this will be examined subsequently.

C. Connection to neural networks

The models we will use to implement above are very simple. Each question will be characterized by a linear model followed by softmax. Specifically, for question n, assume that there are K_n categorical answers. Then we consider the model

$$f_n(v_m) = W_n v_m + b_n \tag{5}$$

where $W_n \in \mathbb{R}^{(K_n-1)\times d}$ and $b_n \in \mathbb{R}^{K_n-1}$. This is a linear model acting on the feature vector $v_m \in \mathbb{R}^d$ for article m, with an added bias b_n. The output $f_n(v_m)$ is a $K_n - 1$ dimensional vector, which is sent into the softmax function to manifest the probability of the K_n categories. Row k of W_n corresponds to $w_{n,k}$ and component k of b_n is $b_{n,k}$, and we consider $k = 1, \ldots, K_n - 1$. It is emphasized that for K_n possible categorical answers, the number of outputs of affine function $W_n v_m + b_n$ is $K_n - 1$, and then one of the inputs to the softmax is set as zero, representing categorical answer K_n. Specifically, in (1) and (2) effectively $w_{n,K_n} = 0_d$ and $b_{n,K_n} = 0$, where 0_d is a d-dimensional vector of all zeros.

Each question is characterized by its own matrix W_n and bias b_n, shared among all documents, and each document has its own feature vector v_m. The feature vector v_m may be viewed as a *learned* d-dimensional embedding vector for document m.

We may impose ℓ_2 regularization on the rows of W_n and on v_m, as in (4). The learning can be performed using neural network back-propagation, but there is no pointwise nonlinearity. Rather, the model is just $W_n v_m + b_n$, with no pointwise nonlinearity imposed, and then this is sent directly into the softmax.

The data we will use to learn these parameters will be provided by the LLM, using our questions and the large number of corresponding articles. We may use tools like Tensorflow to implement the gradient-descent learning. Once we have inferred the feature vectors $\{v_m\}$ associated with all articles, an interesting thing would be to embed them in a 2D space, using a tool like t-SNE. We may see clusters of similar/related documents manifested. We could provide article summaries via the LLM, to various portions of this embedding space, to provide the user with an understanding of what the article landscape looks like.