January 03, 2019

A DEEP GAUSSIAN MIXTURE MODEL APPROACH TO SCORING BASED ON FEEDBACK

Pranjal Daga

Antonio Nucci

A DEEP GAUSSIAN MIXTURE MODEL APPROACH TO SCORING BASED ON FEEDBACK

AUTHORS:
Pranjal Daga
Antonio Nucci

ABSTRACT

Techniques are described herein for taking into account actual written feedback provided by customer in terms of comments to generate a score. The score, in turn, may help in efficient executive decision making. The textual feedback is more holistic as it describes in detail why a customer might dislike/like/love a product.

DETAILED DESCRIPTION

Feedback from customers is at the core of products and services, whether at the ideation, development, optimization, or growth phase. With the rapid realization of the importance of customer feedback, companies have started giving more thought to customer satisfaction metrics like Net Promoter Score (NPS), Customer Health Score (CHS), Customer Satisfaction Score (CSAT), Customer Effort Score (CES), etc. Based on these metrics, when a matrix of customer Identifier (ID) and score is prepared, it is found that the resulting matrix is quite sparse. To address this problem of data sparsity the actual written feedback provided by the customer in terms of comments is taken into account. This textual feedback is more holistic as it describes in detail why a customer might dislike/like/love the product. The score generated via processing the feedback text can augment decision making.

However, there are a couple of issues associated with this textual feedback. The first issue involves identifying pertinent keywords/stances/features from the text to portray customers as well as products. The second issue involves using those features to generate a score for each customer feedback.

To handle the above-mentioned problems and others encountered in prior art, techniques are provided for using Gaussian Mixture Models (GMMs) infused with a deep learning architecture.

To be able to model customer feedback as a score for decision making, it is important to overcome the issues of finding customer characteristics and also identify from the text significant keywords. As described herein, a deep GMM is described to resolve these issues. The architecture is a parallel network on learning both product characteristics and customer personas. GMMs are utilized as they are capable of learning and emulating the scoring of products to customers.

Deep GMM is a joint modeling scheme for both products and customers. A parallel deep learning architecture is augmented with a mixture of Gaussian layer to occupy the communication between products and customers in order to learn the score and weight over various factors. Furthermore, deep GMM utilizes word embeddings to express words, as against traditional bag-of-words techniques in current topic modeling efforts.

Given N samples, a tuple (Prod, Cust, $Score_{prod,cust}$, $E_{prod,cust}$) is provided where

$$prod \in \{1, 2, \ldots, NUM\_PRODUCTS\}$$

represents the list of products, and

$$cust \in \{1, 2, \ldots, NUM\_CUSTOMERS\}$$

represents the list of customers.

$Score_{prod,cust}$ represents the feedback score provided by customer "cust" for product "prod". $E_{prod,cust}$ represents the actual feedback written by customer "cust" for product "prod" which describes in detail (in free form text) whether customer "cust" likes product "prod" or not and the reason for that choice. To simplify, E_{prod} is used to describe all feedbacks (concatenated) written for product "prod" and E_{cust} describes all feedbacks written by customer "cust," concatenated together. The objective is to use this architecture for mapping feedback to a score and then evaluate the score for non-scored feedback.

Figure 1 below illustrates a parallel network architecture to extract features from products and customers separately. Since the architecture is inspired by the Convolution Neural Network (CNN) architecture, the networks are called ConvGMMs. In this architecture, one network $ConvGMM_{product}$ is designed to model product characteristics and the other $ConvGMM_{cust}$ is attributed to model customer behavior. Words are mapped as word embeddings before being fed into some common layers of a CNN model. The CNN model is used to determine feature abstractions on various levels. The network includes a convolution layer, followed by a max-pooling layer and a fully-connected layer.

To converge the two parallel networks, a mixture of Gaussian layer is augmented which helps in reproducing parameters of the mixture model, mean (denoted by Greek letter 'mu') and mixture proportions (denoted by Greek letter 'theta').

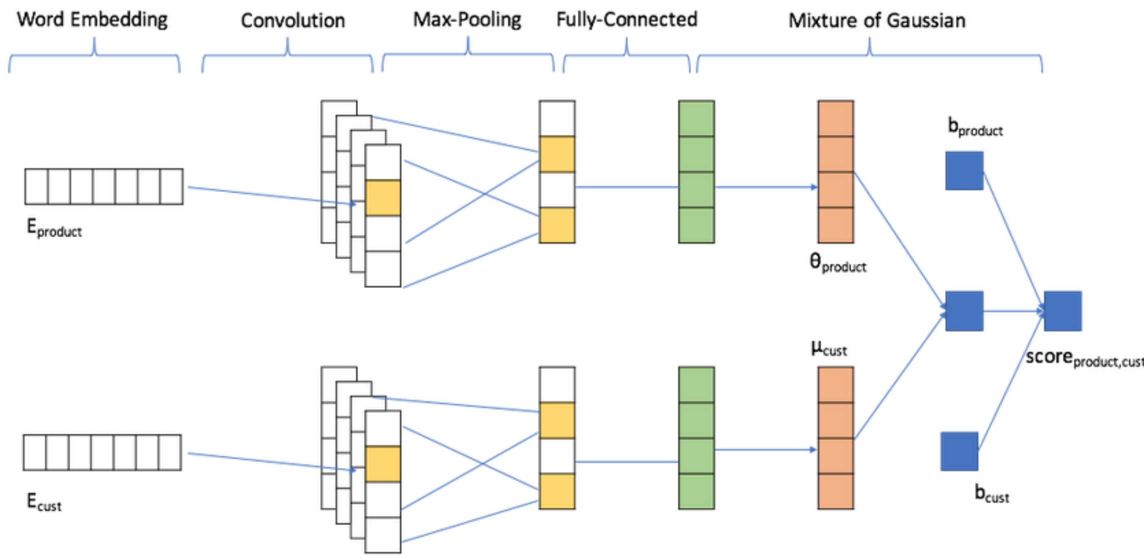

Fig. 1: Architecture Diagram of the proposed Deep Gaussian Mixture Model

Instead of using one-hot encoding wherein the number of dimensions is the entire vocabulary and only one cell is active per token, word embeddings are utilized. Word embedding is a way to reduce the dimensions of the mapped token while capturing more syntactic and semantic information among different tokens. This gives word embeddings an edge over the regular bag-of-words in terms of computation and also captured information. $x_i^p \in \mathbb{R}^D$ represents a D-dimensional word embedding corresponding to i-th token of the p-th product feedback. These embeddings may be concatenated to create the entire product feedback vector. The product feedback vectors are all mapped to equal lengths via padding (wherever required).

To produce a new set of features, a CONV layer applies filter $w \in \mathbb{R}_s^d$ in a sliding window of 's' words. If feature z^i_p is generated from the p-th product via a sliding window of length $x^p_{i:i+s-1}$, then it would look like this:

$$z_i^p = relu(w^p * x_{i:i+s-1}^p + b)$$

Rectified Linear Unit was chosen as the activation function to have sparsity and reduced likelihood of vanishing gradients.

Out of this sliding window, the maximum value may be selected as features by using the MAX_POOLING operation.

$$o = max(z^p)$$

Finally, the final output (i.e., product embedding) is obtained by applying the FULLY_CONNECTED layer:

$$h^p = relu(W^p x O^p + b^p)$$

And similarly for customers, it becomes:

$$h^c = relu(W^c x O^c + b^c)$$

To capture the relationship between customers and products, a mixture of the Gaussian layer was designed on top of the FULLY_CONNECTED layer. This is instrumental in considering the importance of each factor in determining the overall score. Thus, based on product and customer embeddings, the FULLY_CONNECTED layer is utilized to stimulate (θ, μ).

And so the parameters of the GMMs become:

$$\theta^p = l^p(W^{Gaussian} * h^p + b^{Gaussian})$$

and

$$\mu^c = l^c(W^{Gaussian} * h^c + b^{Gaussian})$$

Combining everything, the objective function of the deep GMM becomes

$$L = \sum_{c=1}^{c=N_c} \sum_{p=1}^{p=N_p} \left\| \sum_{k=1}^{k=K} \mu_c^k * \theta_p^k + b_c + b_p - score_{c,p} \right\|$$

where N_c is the total number of customers and N_p is the total number of products, and b_u and b_p are the bias of customer and product respectively. This equation is only valid if $score_{c,p}$ is observed in training data. The entire network is trained by minimizing the objective function. The model is optimized via Stochastic Gradient Descent and in order to prevent overfitting. Regularization is achieved with the help of dropout on the first layer of the network.

Techniques described herein use parallel deep learning architecture to allow one of the two neural networks to just model the customer's choices. Using raw feedback text may enable understanding not only the product but also the customer. Furthermore, diving deep into modeling both customers and products may help generate a comprehensive score which can be used for decision making. GMMs are capable of learning and emulating the scoring of products to customers.

In summary, techniques are described herein for taking into account actual written feedback provided by customer in terms of comments to generate a score. The score, in turn, may help in efficient executive decision making. The textual feedback is more holistic as it describes in detail why a customer might dislike/like/love a product.