Introduction

→ Conduct rapid experimentation

→ stronger signal on the breadth of the people’s interest

→ Computationally efficient way to ensure that our recommendations are of high quality

Optimal recommendation choices

→ long term interest

→ recent content

→ Multiple recommendations algorithm in a principle way,

Large data and how to better the scrolling

We use the top-ranked posts from the distillation model as the ranking candidates for the later-stage high-performance ranking models.

Setting up the distillation model’s mimicry behavior minimizes the need to tune multiple parameters and maintain multiple models in different ranking stages. Leveraging this technique, we can efficiently evaluate a bigger set of media to find the most relevant media on every ranking request while keeping the computational resources under control.

the candidate generation stage (also known as sourcing stage) and the ranking stage.

Different components for precision mental health recommendation

1. Data
   1. Impression
   2. Positive signal
   3. Negative signal
2. Objective Function - CTR
3. Logging
4. Features
   1. Dense Features : users/items pairs, historical statistics, predictions from upstream models
   2. Sparse Feature : Typically sparse features include IDs of users/items, demographics, keywords etc.
   3. How are they learned, ranking, ordering is important
      1. or example, the relative ordering of dense feature values are meaningful but this is rarer for sparse features.
      2. A multilayer perceptron (MLP) transforms dense features into a vector with a predefined dimension. Different from dense features, sparse features are mapped into numerical vectors via embedding lookup. A dot product of vectors is used in order to calculate the second-order interactions between dense and sparse features. Because of this the vectors that represent sparse and dense features must share the same dimensionality. Both the vectors generated from dense and sparse features, as well as second-order interactions, are fed into a downstream MLP to predict the relevant targets.
      3. A major difference is that the normalized dense features are directly concatenated with the sparse feature embeddings rather than being processed by a separate complex MLP. The concatenated vector, which includes both the normalized dense features and the sparse feature embeddings, are fed into a downstream MLP with 2 hidden layers for prediction.
      4. The values of dense features may fall into some arbitrary range, unlike sparse features whose values are typically restricted. An arbitrary range leaves open the possibility of outliers which can significantly impact both the training and serving of neural network models. For example, an unusually large value can make a sigmoid neuron saturated and thereby block the information of other values from passing through this neuron. To alleviate this issue, Airbnb uses the z-score of feature values when their distribution is approximately Gaussian. For power law distributions they divide the feature values by the corresponding median and perform a log transformation. This distribution specific normalization works well in Airbnb’s environment according to their experimental results. Two drawbacks are that outliers can still appear after the normalization and that this approach will not always handle an arbitrary distribution
      5. The underlying assumption of Google’s approach is that the prediction targets are correlated with the relative ordering of feature values. This assumption is correct for many cases though the correlation is not necessarily linear. For example, the expected click-through rate (CTR) of a post can be nonlinearly correlated with user’s age. CTR often increases with users’ age at the beginning, peaks at a certain age and drops down after the age. In contrast, quantiles of feature values are monotonic. For the previous example, normalized age values are increasing with raw age values. The correlation between expected CTR and normalized age values is still non-linear. Activation functions for MLPs such as ReLU are often piecewise linear. The downstream MLP may need a few layers to learn the complex non-linear relationships between the prediction targets and the ordering of feature values.
      6. In order to avoid the additional complexity we propose to map a dense feature value into an integral ID based on the corresponding quantiles. More specifically, quantiles are precomputed according to historical feature values and used to define the boundaries of a sequence of bins. Each bin is assigned with a unique ID. Each feature value is mapped to the ID of the bin to which this value belongs. As each dense feature value can only belong to one bin and the total number of bins for a dense feature is typically large, the set of IDs generated from all dense features for an instance can be represented by a sparse binary vector. If IDs of dense and sparse features are included in a global ID dictionary, these IDs can be used to look up embeddings from a global embedding matrix which allows for dense and sparse features to be processed in a unified way. The embeddings of both dense and sparse features are fed into a downstream MLP for prediction. We call this approach quantile sparsification (Figure c). We developed and launched quantile sparsification earlier this year. Recently, Netflix published a similar technique to map dense features to one-hot vectors according to quantiles [5]. Their technique was shown to be effective according to their experimental results.
      7. At Quora, embeddings of quantile bins rather than the quantiles themselves are used for prediction. Embeddings of quantile bins are trained independently and can easily adopt the linear or non-linear relationships between ordering of feature values and the prediction targets. The downside to this approach is that models with quantile sparsification are more likely to overfit data since the number of embeddings significantly increases. Regularization on these embeddings is essential. For example, techniques like Tikhonov regularization can be used for reducing differences between embeddings of adjacent quantile bins. This is because the corresponding embeddings of two adjacent quantile bins represent similar dense feature values and thereby these two embeddings should be similar to each other in most cases.
      8. In conclusion, different companies have successfully adopted different approaches to the unification of dense and sparse features. The most common approaches are to normalize dense features or feed them to a separate MLP and to process sparse features using embedding lookups. The approach we call quantile sparsification transforms dense features into a sparse form similar to other sparse features which unifies dense and sparse features while reducing the architectural complexity of our neural network models.

Deep learning based recommendation systems

1. <https://arxiv.org/pdf/1906.00091v1.pdf>

Notes :

1. Catastrophic forgetting

References

1. <https://www.nvidia.com/en-us/glossary/data-science/recommendation-system/>
2. <https://cloud.google.com/retail/docs/models#model-types>
3. <https://builtin.com/data-science/recommender-systems>
4. <https://www.altexsoft.com/blog/recommender-system-personalization/>
5. <https://www.itransition.com/machine-learning/recommendation-systems>
6. https://quoraengineering.quora.com/Unifying-dense-and-sparse-features-for-neural-networks