Layout of Experimental Method for Comparison and Evaluation of Topic Modelling Algorithms in Information Filtering

Final Year Project

Jack Power
20080169
Supervisor: Rob O'Connor
BSc (Hons) in Applied Computing (IoT)

Aim

The aim of this experiment is to compare and contrast different approaches to constructing a topic model based on some body of documents. Various algorithms yield different results, aside from raw metrics, the topic space generated can also have special properties. An example of this is the word2vec algorithm, when sufficiently trained the model naturally represents word associations through vector similarity. E.g. The vector representation of “King” is to “Man” as that of “Queen” is to “Woman” (Mikolov et al, 2013).

A commonplace example of where such topic modelling is useful is in recommender systems, such as there are on many media streaming services. Amazon SageMaker provides the object2vec algorithm, a generalisation of word2vec which is capable of embedding generic objects, preserving semantic similarity in the same fashion as with text (AWS, 2021).

Such a model trained on a list of movies for example, along with associated ratings for each user, would be capable of predicting how a user would score an unseen movie based on the latent semantics extracted from the data.

This technique of predicting the rating of a single user based on the previous ratings of others in known as collaborative filtering. A high-profile example of such a recommendation task is the Netflix Prize (2009). From 2006 to 2009 Netflix held an open competition for teams to submit algorithms which would score better than their own given a dataset of movie ratings. Each rating contained only a numerical user and movie ID, a rating from 1 to 5 and the date of the rating. The portion of the dataset used for training consisted of over 100 million ratings from 480,000 users for 17,000 movies. The metric used to score the predictions is RMSE (root-mean-square error), that is the absolute deviation of the prediction from the true rating which was unknown to the model. The algorithm which ultimately won was BellKor’s Pragmatic Chaos which achieved an RMSE of 0.8567, an 10.06% improvement over Netflix’s own Cinematch (0.9525).

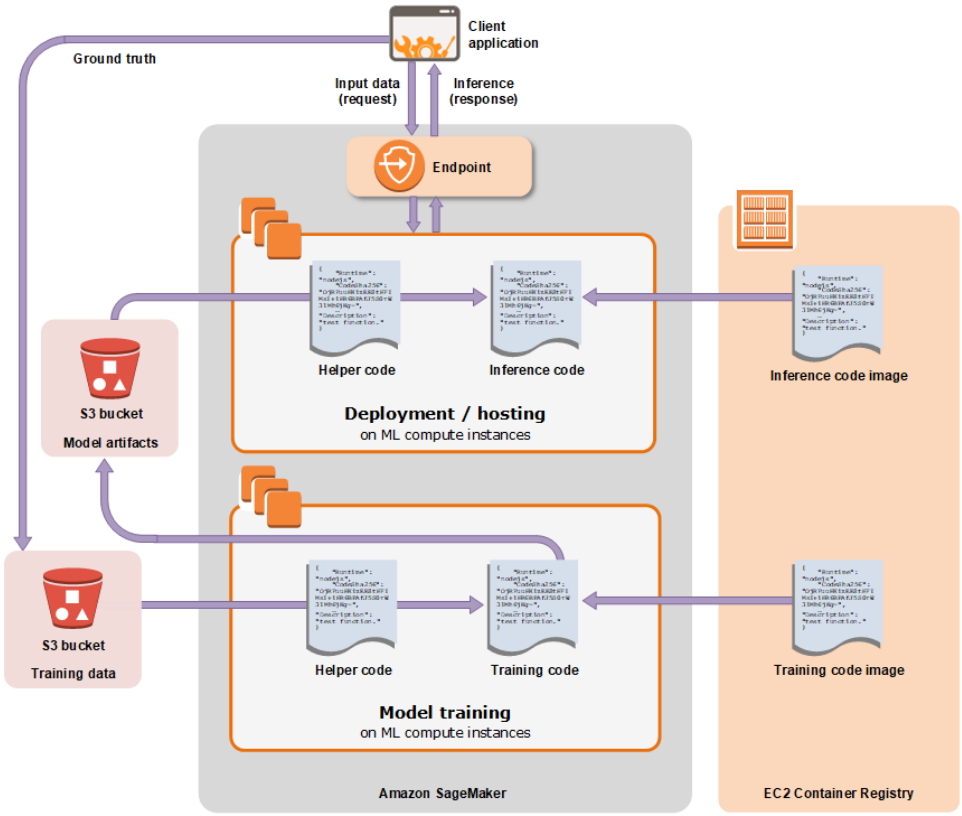
For this experiment the same metric will be used as it is exceedingly common and provides easy comparison between different algorithms and implementations. The dataset to be used is the MovieLens 100K dataset, 100,000 ratings from 1000 users for 1700 movies (GroupLens, 2021). This is a well-known dataset, and should prove much more manageable than one as colossal as the Netflix dataset.

Method

In order to construct an experiment of sufficient rigor and repeatability, a scientific testbed will be used in the form of Amazon SageMaker, in particular making use of the Experiments feature. SageMaker Experiments organise the data and algorithms that were used in the training of a model so that trials may be easily compared and evaluated. Each experiment may have many trials which encapsulate the inputs, parameters, configurations and results for each particular training run of the model (AWS, 2021).

Aside from the organisational capabilities, SageMaker Experiments also provides tooling which allows for real-time data visualisation as the experiment progresses. The generated metric charts and graphs should prove invaluable in the presentation of the final results.

Shown is a diagram illustrating the training and deployment of a model using SageMaker (AWS, 2021).



The training code image is first loaded into an ML compute instance and trained on the data provided via an S3 bucket. There are a range of high-performance algorithms built into SageMaker which greatly simplifies their use. It is also possible to provide custom algorithms which may be dependent on any of the machine learning frameworks which SageMaker supports (TensorFlow, PyTorch, Apache Spark, etc.).

The generated model artifacts are passed into a destination bucket which constitutes the trained model. The inference code image is then loaded which exposes the model for querying via an endpoint. Shown at the top is a client application requesting inference (a prediction) based on some unseen data. In our case this could be asking for a predicted rating for a given movie and user ID, as would be done to validate the model.

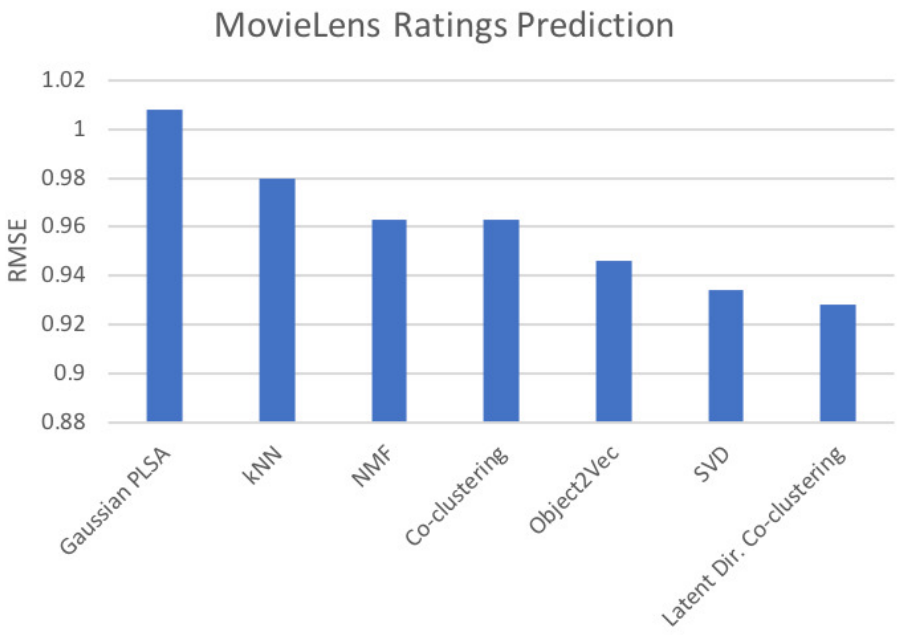
Another way of utilising the generated vector space is in the context of a nearest neighbours problem. Since our model has been trained to recognise the latent features which dictate how certain users rate certain movies, movies which are similarly rated by users with similar tastes ought to be similar movies.

Hypothesis and Results

The experiment will be constructed based upon an example notebook supplied by Amazon (AWS, 2019). In it the object2vec algorithm is trained on the MovieLens 100K dataset, and the resultant model's metrics compared to alternative algorithms.

As discussed previously, SageMaker provides facilities to encapsulate model training parameters for easy comparison. The training job which yielded the results below was achieved using a relatively modest configuration for the sake of speed. The epochs used, that is the number of times the dataset was processed in its entirety, was 20. SageMaker also provides Hyperparameter Optimisation which trains the model multiple times to determine the optimum configuration. Although an extremely powerful tool, the computation involved makes it equally costly.

Shown is the supplied chart which compares the computed RMSE error metric for object2vec to those declared by the various libraries which provide the other algorithms.



Note the presence of PLSA (Probabilistic Latent Semantic Analysis), SVD (referring to the Singular Value Decomposition of Latent Semantic Analysis) and Latent Dirichlet Co-clustering (Latent Dirichlet Allocation). These techniques have all been previously identified in the course of this project as areas of further exploration.

It is planned to expand on the example provided by implementing these algorithms in much the same fashion as has been done with object2vec. AWS SageMaker Experiments will be used to organise and validate the training of the models. The results generated will be rigorously compared and the presentation of the data will serve to illuminate the nature of these algorithms. It is hoped that the complete experiment will be packaged so that it may be run automatically by any who wish to generate and explore the results for themselves.

References

Mikolov, T., Chen, K., Corrado, G., Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. *The International Conference on Learning Representations*.

AWS, (2021). *Object2Vec Algorithm - Amazon SageMaker*. [online] Available at: https://docs.aws.amazon.com/sagemaker/latest/dg/object2vec.html [Accessed 2 Mar. 2021].

Netflix Prize, (2009). *Netflix Prize: Review Rules*. [online] Available at: https://netflixprize.com/rules.html [Accessed 2 Mar. 2021].

GroupLens, (2021). *MovieLens 100K Dataset*. [online] Available at: https://grouplens.org/datasets/movielens/100k/ [Accessed 2 Mar. 2021].

AWS, (2021). *Manage Machine Learning with Amazon SageMaker Experiments*. [online] Available at: https://docs.aws.amazon.com/sagemaker/latest/dg/experiments.html [Accessed 2 Mar. 2021].

AWS, (2021). *Train a Model with Amazon SageMaker*. [image] Available at: https://docs.aws.amazon.com/sagemaker/latest/dg/how-it-works-training.html [Accessed 3 Mar. 2021].

AWS, (2019). *object2vec movie recommendation*. [online] Available at: https://github.com/aws/amazon-sagemaker-examples/blob/master/introduction_to_amazon_algorithms/object2vec_movie_recommendation/object2vec_movie_recommendation.ipynb [Accessed 3 Mar. 2021].