Romance or Thriller?

A Movie Genre Classification Problem

Anonymous – COMP90044, Semester 1, 2020

Movies have become an integral part of the modern entertainment industry, and one of the key features of a movie is the movie's genre. The genre of a movie describes stylistic properties of a movie, such as the tone or the mood of the movie and can be a deciding factor as to whether an audience will choose to watch a movie or not. Being able to automatically assign these genres may become beneficial for entities such as movie hosting sites as their library becomes large.

It is my hypothesis that models that can learn complex linearly non-separable data sets will perform well in this problem. In this paper, I will show two candidate machine learned models as solutions to this classification problem, with results showing up to 45% accuracy with only a small amount of model tuning, utilising both candidate classifiers.

Previous Work

The primary use of the dataset used in this paper is in the sub field of recommender systems, wherein the problem to solve is to build a system that attempts to predict a user's preference or rating towards an item, in this case a movie. The techniques that have been explored include the use of "deep learning" (Deldjoo et al, 2018b) in recommender systems.

Other work exists that relates to genre classification specifically, also. Many approaches to this problem attempt to solve a more general multi-label problem, as this better reflects the real world with movies often falling into multiple genres. This work uses a variety of models and datasets. For instance, there has been the use of deep neural networks to classify movies based on their movie posters (Chu et al, 2017), and novel architectures based on convolutional neural networks and temporal information extracted from image-based features (Wehrmann et al, 2017).

Dataset and Features

The data used in this paper is derived from the MMTF-14K (Deldjoo et al, 2018a) and MovieLens (Harper, 2015) datasets. The features used from this larger dataset fall into three categories, metadata, audio and visual. The metadata features include the movie's title, a list of user generated tags (such as "dark" or "humorous") and the year in which the movie was released. For the purposes of this paper, the audio and visual features are treated as a black box, though they can be thought of as some numerical value which encapsulate some feature about the movie, such as the lighting or the pitch of the sounds.

The data is split into (predefined) training and validation sets, of size 5420 and 299, respectively. These sets are used in the evaluation of the models using a hold-out method after pre-processing.

Most features are pre-processed in a manner that is appropriate to their type.

The audio and visual are scaled using a min-max scaler, first by fitting the scaler with the training set, and then separately transforming training and validation data separately.

The textual features (title and tag) are first case normalised (lowercase) and Unicode elements, such as character accents are stripped, normalising the data. From here, a term-frequency inverse document-frequency (TF-IDF) transformation is applied, first by fitting the transformer to the training class and then transforming the training and validation data sets separately.

Unfortunately, after pre-processing, the title column spawns many, sparsely populated, features because the words in titles vary drastically. I have chosen to exclude these features from further experiments, as preliminary results showed no appreciable change in accuracy upon the inclusion of the title features.

This separation of the training and validation in the pre-processing steps in this is key to not allow any information of the validation set to leak into the training set.

Models

The two baseline models used for comparison in these experiments are the zero-rule classifier and the one-rule classifier (in the form of a decision tree with maximum depth of 1). These two were chosen as they are used broadly as a standard baseline in machine learning contexts and provide insight into what extremely simple models can achieve. This also provides some insight into the difficulty of the problem, as high performance with these models indicates that the problem may be relatively simple, as well as insight into the skewness of the data.

Further, two candidate models were chosen in attempt to classify movies' genres. These are a multiplayer perceptron classifier (MLP) utilising a feed-forward network, a support vector machine classifier using a one-vs-rest multiclass classification scheme (SVM).

As a reference to the ability of models that are not designed to work on linearly non-separable data, a Perceptron (P) is also tested. This model is also used out the box.

Both candidate models were chosen for their ability to learn complex, linearly non-separable, data sets, as it is my suspicion with this data set that it is linearly non-separable due to its high dimensionality. The MLP classifier was also chosen as there is frequent use of other neural-network models in similar contexts (Chu et al, 2017; Wehrmann et al, 2017). The SVM classifier was chosen as it is used as a comparable model to the MLP in some contexts (Zanaty, 2012).

Various hyper-parameters have been tuned for each of the two candidate models to find some optimal performance of these models, for comparison. The baseline models' performance is evaluated on out-of-the-box parameters.

Model Analysis

The two baseline models performed similarly in the single test performed with each model. The zero-rule's chosen class is the "Romance" class, which is the majority class for this data set. The single decision node in the tree forms a splits on the value of the feature `tag_sci` at a cut-off value of 0.05, then classifies instances as "Romance" and "Sci_Fi". The effect of choosing to use a small number of labels for the whole data set skews the predictions towards whatever labels are chosen. This effectively produces a large recall value for the chosen classes, while scoring zero for all other, leading to poor macro recall scores and mediocre weighted recall scores. The precision scores for these models are generally much worse than their recall. As the predicted classes do not form a majority, their FP rate is relatively high, leading to lower precision scores. As both metrics do not score well as the macro and micro scale, the F-1 score is also appropriately low. The accuracy is 17% for the zero-rule classifier and 19% for the one-rule classifier.

The hyperparameters of the two candidate models have been tuned using a grid search over a small parameter space. These hyperparameters are likely not the optimal parameters for these models, though tuning these parameters too much is a form of over fitting and will not always produce the best results for all data sets.

For the MLP, the hyper parameters tuned are the network configuration (number and size of the hidden layers), the solver used in the training process (Adam, SGD, LBFGS), the activation function used (identity, logistic, tanh, and ReLU), the learning rate (constant, inv-scaling, and adaptive) and the number of epochs.

For the SVM, the tuned hyper parameters are C a regularisation parameter, the type of kernel used (polynomial, sigmoid, RBF), the degree of the non-linear kernels, and the kernel coefficient γ (scaled or unscaled).

	Accuracy	Recall		Precision		F-1	
		Weighted	Macro	Weighted	Macro	Weighted	Macro
MLP	0.45	0.45	0.31	0.45	0.32	0.42	0.30
SVM	0.45	0.44	0.37	0.45	0.38	0.43	0.36
P	0.30	0.30	0.31	0.46	0.33	0.29	0.25
0R	0.17	0.19	0.08	0.06	0.04	0.05	0.08
1R	0.19	0.17	0.06	0.03	0.01	0.05	0.02

Table 1: Model Performances Utilising Tuned Hyperparameters. Note: The macro scores underrepresent the models' performance as there is a heavy skew in the sizes of each class.

Using a grid search, the optimal hyper parameters found for the MLP classifier to be a network configuration of $(80,)$, using the SGD solver with a constant learning rate and a ReLU activation function. For the SVM classifier, the optimal hyper parameters are a value of $C = 10$, a cubic (degree 3) polynomial kernel, and a scaled γ.

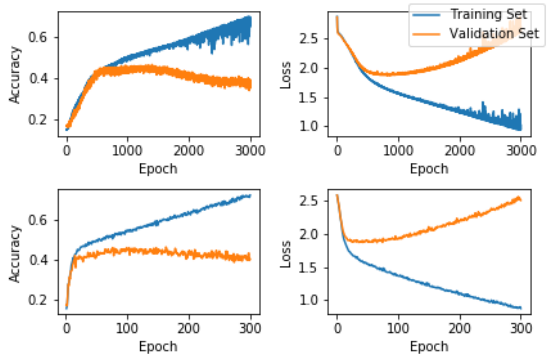

Figure 1: Accuracy and Loss vs Epochs. SGD solver (top), Adam solver (bottom).

Interestingly, the behaviour of the solver used for the MLP greatly affects how many epochs are required to fit the model. The two scorers that produced the highest scoring classifiers, Adam and SGD, thought the number of iterations required to reach the optimal validation accuracy was much lower with the Adam solver. The Adam solver took on average 100-125 epochs before the accuracy began to drop, while the SGD solver exhibits similar behaviour at an average of 1250 epochs. After both of their respective number of epochs for each solver, the loss on the validation set no longer follows a similar downwards trend as with the training set, and instead starts to increase. This overfits the model, which accounts for the decreasing accuracy.

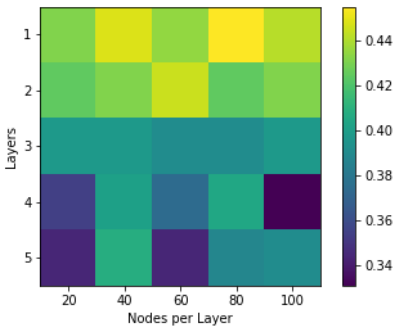

Figure 2: The accuracy of various network configurations of the MLP.

Further, the size of the neutral network plays an important role in the performance of the model. There is a tendency for the MLP to perform better when the number of hidden layers is small (i.e. 1 or 2), and there is less effect by the actual number of neurons utilized in the layers. Increasing the number of layers beyond 2 greatly diminishes the model's ability to learn the data set, producing an underfit model with a much lower accuracy score on the validation set.

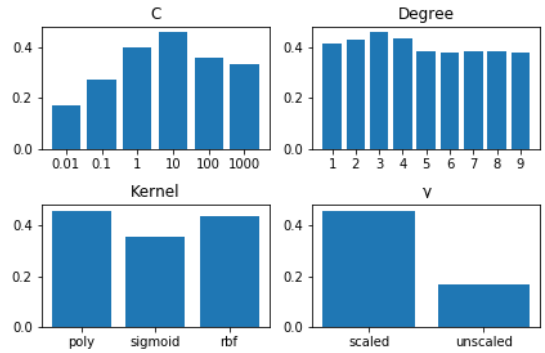

Figure 3: SVM Accuracy vs various parameters. Where parameters are unspecified, they are C=10, degree=3, kernel=poly γ=scaled.

The SVM classifier's C parameter controls how the model handles how the margin is fit between classes. Higher values of C, in general, lead to lower bias and higher variance. Finding an optimal value, such as the value found of 10, will lead to a good trade-off between variance and bias. A lower C value fits a "soft margin" where the margin allows for some instances in the training set to be mislabelled, however this will create a larger margin in general. A higher value of C will cause the training set to be more correctly labelled at the cost of creating a tighter margin.

The degree of the polynomial kernel is also important for the behaviour of the SVM classifier. A higher degree will cause he margin to have a more complex shape, though this can cause overfitting as shown with this classification problem, where values that are too large perform worse than smaller values. As a linear polynomial does not perform well, this implies the data set is not linearly separable.

The type of kernel affects the model's performance. The polynomial kernel performed best overall, though performs similarly well to the RBF kernel.

The γ coefficient also affects the bias-variance trade-off balance. Lower γ values tend towards lower bias and higher variance, with higher values increasing the bias and decreasing the variance. The γ parameter was chosen optimally to be "scaled" (lower) meaning this contributed to increasing the model's variance and decreasing the bias.

Conclusion

Overall, the performance of the two candidate models were similar, with the SVM scoring slightly higher in all tested metrics excluding macro-recall (this value, however, is only less by 0.01. Both models outperformed the baseline models, showing that they are both adequate candidates for solutions for this problem. My hypothesis that models that are designed to learn linearly non-separable data would perform well on this task was validated. However, other models, such as the Perceptron also outperformed the baseline models. With further hyper-parameter tuning the performance of these models can likely be enhanced.

References

Deldjoo, Yashar and Constantin, Mihai Gabriel and Schedl, Markus and Ionescu, Bogdan and Cremonesi, Paolo. (2018a). MMTF-14K: A Multifaceted Movie Trailer Feature Dataset for Recommendation and Retrieval. Proceedings of the 9th ACM Multimedia Systems Conference, MMSys 2018, Amsterdam, The Netherlands, June 12-15, 2018

Deldjoo, Y., Elahi, M., Quadrana, M., & Cremonesi, P. (2018b). Using visual features based on MPEG-7 and deep learning for movie recommendation. International journal of multimedia information retrieval, 7(4), 207-219.

Chu, W. T., & Guo, H. J. (2017). Movie genre classification based on poster images with deep neural networks. In Proceedings of the Workshop on Multimodal Understanding of Social, Affective and Subjective Attributes (pp. 39-45).

Wehrmann, J., & Barros, R. C. (2017). Movie genre classification: A multi-label approach based on convolutions through time. Applied Soft Computing, 61, 973-982.

Zanaty, E. A. (2012). Support vector machines (SVMs) versus multilayer perception (MLP) in data classification. Egyptian Informatics Journal, 13(3), 177-183.

F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4, Article 19 (December 2015).