Box Office Prediction for Upcoming Films
Final Report

Rui Xue, Yanlin Chen

1 Introduction

In this project, we apply machine learning algorithms including multiclass Naïve Bayes and SVM to predict box office for movies. In our model, the predicting problem is converted into a multiclass classification problem-rather than predicting the exact value of box office. Firstly, we collect movie information from the internet, and use Naïve Bayes as our prototype model for training and testing data. Given the feedback, we improve our feature selection by varying the training data size. Lastly, we compare two different techniques, Naïve Bayes and SVM, giving birth to the final feature and model. The method and results are detailed in the following sections.

2 Data Collection and Feature Generation:

2.1 How we get the original data

We write some python script to grab all the US film information from 2010 to 2011 from Internet Movie Database (a.k.a. IMDB, http://www.imdb.com). And there are more than 7,000 films and only around 409 films have some box office income. So we choose those 409 films as our training & testing set.

2.2 Features for upcoming films

From the IMDB, for each film, there are a lot of features, but for an upcoming film, the number of features is quite limited. Here are the features we choose.

Important Features: Director, Writer, Actors, Genre, Release Date, Estimated Box Office (Yes! There is estimated box office, but it has never got the correct estimation), Production Company.

Less Important Features: Title, Aspect Ratio, Color Company, Duration, Language, Filming Location, Sound.

Useless Features: Meta Score, News Count, Photo Count, Rating, Rating Count, Review Count, Video Count.

The useless features are mainly because it is either untreatable (e.g. rating and reviews) or they are hard to get as training data. For example, it is impossible to know the news count before the film “The Social Network” was released NOW. So we will not use the Useless Features in our machine learning progress.

2.3 Feature Generation

The types of the input features include contiguous values and discrete values. The main idea is to reduce the number of values for each feature. And we also drop some of the features that all the films almost share the same value, such as Color Company and Aspect Ratio.

Here is the table of each feature input and how we generate final training & testing features.

Feature	Type	Number of values	How to reduce the number of value	Final Output number of values
Directors	Discrete	500+	Use the number of Google search result to classify how famous this name is. (famous, normal, not famous)	3
Writers	Discrete	500+		
Actors	Discrete	1000+		
Production Company	Discrete	400+		3
Genre	Discrete	21		21
Release Date	Discrete	365 * 2	We use months (Jan - Dec)	12
Estimated Box	Contiguous	Infinite	(same as box office classification in part 4)	10
Duration	Contiguous	100-	(very long, long, normal, short, very short)	5

Note: We merge the directors, writers and actors to be the names feature to reduce the number of features.

3 Method

To solve this classification problem, we consider using two major method including 1) multiclass Naïve Bayes and 2) Multiclass SVM.

3.1 Multiclass Naïve Bayes:

D-class Naïve Bayes have the similar model as the binary one:

$$\arg\max_{y_i}\left(\prod_{j=1}^{n} p(x_j \mid y=1)\right)p(y=y_i),\quad y \in \{y_1, y_2, \dots, y_D\}$$

We apply the multiclass Naïve Bayes algorithm with Laplace smoothing to train the data, and obtain the parameters using maximum likelihood estimates as the following formula:

$$\forall y_l \in \{y_1, y_2, \dots, y_D\},\quad \phi_{tk|y=y_l} = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n_{ti}} 1\left\{x_{tj}^{(i)} = k \wedge y^{(i)} = y_l\right\} + 1}{\sum_{i=1}^{m}\{y^{(i)} = y_l\} n_{ti} + |V_t|}$$

where t indicates the t^{th} type of feature, each feature has k dimensions and $|V_t|$ is the dimension of t^{th} feature. And similar to binary Naïve Bayes,

$$\phi_{y=y_l} = \frac{\sum_{i=1}^{m} 1\{y^{(i)} = y_l\}}{m}$$

3.2 SVM

Super Vector Machine is inherently binary classifier, however, it can be extended for multi-class classification. The two major strategies, "one against all" and "one against one", both try to decompose the multi-class problem into several two-class sub-problems, and use standard SVM to solve each binary problem. With the result from the paper by C.-W. Hsu and C.-J. Lin, 2002, we choose one against one to decompose our problem. Assume that we have k classes. The one against one method constructs k(k-1)/2 classifiers, each of which is trained on data from two classes. Say the ith and the jth class is chosen to construct the SVM, then we have to solve the following problem:

$$
\begin{aligned}
&\min_{w^{ij},b^{ij},\varepsilon^{ij}} \frac{1}{2}(\mathbf{w}^{ij})^T \mathbf{w}^{ij} + C\sum_{t=1}^{m} \varepsilon_t^{ij} \\
&\text{s.t.} \quad (\mathbf{w}^{ij})^T \Phi(x_t) + b^{ij} \geq 1 - \varepsilon_t^{ij}, \text{ if } \mathrm{y}_t = i, \\
&\qquad\ (\mathbf{w}^{ij})^T \Phi(x_t) + b^{ij} \leq -1 + \varepsilon_t^{ij}, \text{ if } \mathrm{y}_t = j, \\
&\qquad\ \varepsilon_t^{ij} \geq 0, \text{ for i = 1,...,m}
\end{aligned}
$$

where the training example is $\{(x^{(i)}, y^{(i)}); i=1,..., m\}$. The kernel is used to transform data from the input sample to the feature space by kernel function $K(\mathbf{x}_i, \mathbf{x}_j)=\Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_i)$. There four common kernel options:

$$
K(\mathbf{x}_i,\mathbf{x}_j) = \begin{cases}
\mathbf{x}_i^T\mathbf{x}_j, & \text{Linear} \\
(\mathbf{x}_i^T\mathbf{x}_j + 1)^d, & \text{Polynomial} \\
\exp(-\gamma\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^3), \gamma > 0, & \text{Radial Basis Function} \\
\tanh(\kappa\mathbf{x}_i^T\mathbf{x}_j + c), \text{ for some } \kappa > 0 \text{ and } c < 0, & \text{Sigmoid}
\end{cases}
$$

In our model, linear kernel has lower test error rate that any other three common kernels, so linear kernel is used.

4 Feature selection

4.1 Models

Our feature dimension is now reduced to 50 degrees as *Model I*. The features are from directors, actors, writers, company, duration, estimated budget, genre and release date. Firstly, we reconsider the discretization of release date. We reduce the discrete value for release data from years and months to 4 seasons, meanwhile reduce the duration bucket from 5 to 3, obtaining the feature with degree of 44 as *Model II*.

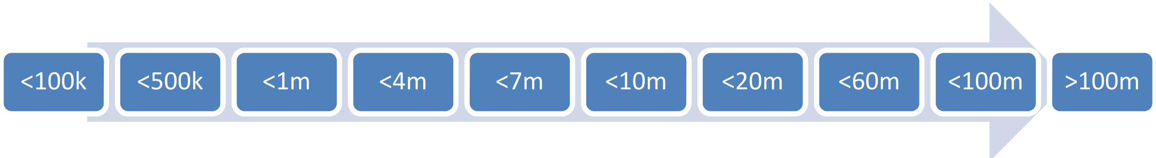
We have the *Model III* with a feature degree of 41 which does even have movie duration as our feature. Lastly, we have *model IV* with a feature degree of 36 that reduce the release to date to only two values, hot time or not hot time, plus reduce the number of bucket for estimated budget.

4.2 Cross Validation

Our data sets (409 movies) are divided into training and test set with each percentage 90% and 10% correspondingly. By using cross validation, we not only experiment on the four models to compare them with each other for feature selection, but also compare Multiclass Naïve Bayes and Multiclass SVM.

4.3 Estimated Box Office Classification

In our experiment, we are solving ten classes classification problem. The ten buckets of estimating the box office is listed in the following figure.



4.4 Error Evaluation

The original error is 1{h(x) != y}. This function needs to be improved. If the actual box office is $1,000,000, then error of a prediction of $10,000,000 is very different from a prediction of $2,000,000. In calculating error, we only allow one class mismatch. For example, if the real value is between 1m-4m, then we think the following three classes are all proper estimation (no error): 500k-1m, 1m-4m and 4m-7m.

5 Results

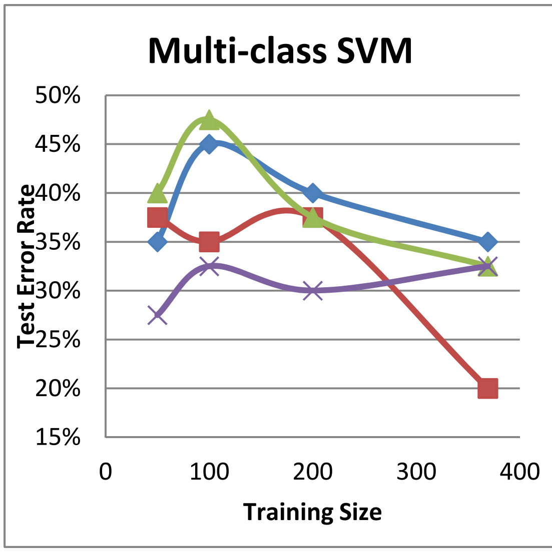
Model I (feature 50)								
	Multiclass Naïve Bayes				Multiclass SVM			
training size	50	100	200	369	50	100	200	369
test error	32.50%	32.50%	30.00%	30.00%	35.00%	45.00%	40.00%	35.00%
training error	8.00%	20.00%	23.50%	27.64%	0.00%	1.00%	5.50%	14.09%

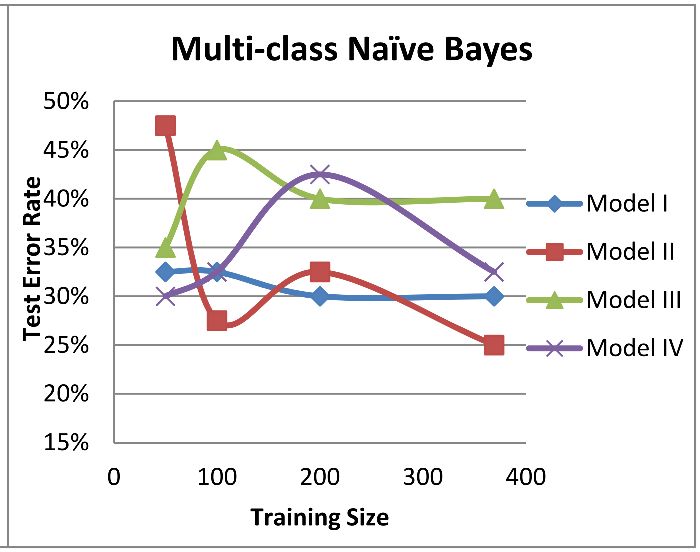
Model II (feature 44)								
	Multiclass Naïve Bayes				Multiclass SVM			
training size	50	100	200	369	50	100	200	369
test error	47.50%	27.50%	32.50%	25.00%	37.50%	35.00%	37.50%	20.00%
training error	22.00%	28.00%	34.00%	31.17%	0.00%	4.00%	16.00%	23.58%

Model III (feature 41)								
	Multiclass Naïve Bayes				Multiclass SVM			
training size	50	100	200	369	50	100	200	369
test error	35.00%	45.00%	40.00%	40.00%	40.00%	47.50%	37.50%	32.50%
training error	40.00%	47.50%	37.50%	32.50%	0.00%	6.00%	16.00%	23.85%

Model IV (feature 36)								
	Multiclass Naïve Bayes				Multiclass SVM			
training size	50	100	200	369	50	100	200	369
test error	30.00%	32.50%	42.50%	32.50%	27.50%	32.50%	30.00%	32.50%
training error	14.00%	22.00%	26.00%	23.31%	6.00%	8.00%	15.50%	18.70%

From the four tables, we can draw the conclusion that Model II overcomes with the rest three in generalizing the test error rate. Secondly, SVM often produces a better estimation that Naïve Bayes.





The above two figures clearly illustrate our choice, *Model II* and SVM as our multiclass classifier. The final feature list:

Feature Type	No. of Values	Possible Discrete Values
Actor/Director/Writer Popularity	3	High, Medium , Low
Company Value	3	High, Medium, Low
Release Date	4	Spring, Summer, Fall, Winter
Genre	21	Action, Adventure, ... , Western
Estimated budget	10	100k - 100m
Duration	3	Long, Medium, Short

6 Prediction for Upcoming Films (in US $)

Movie Name	Prediction Using Naïve Bayes	Prediction Using SVM
The Girl with the Dragon Tatoo	> 100 million	1 million – 4 million
Mission: Impossible – Ghost Protocol	> 100 million	> 100 million
The Adventures of Tintin	> 100 million	4 million – 7 million
The Darkest Hour	10 million – 20 million	10 million – 20 million

And let's see how our prediction works!

7 Acknowledgements

We gratefully acknowledge Professor Andrew Ng and TAs of CS 229's instructions and assistance in our project. We also acknowledge C.-C. Chang and C.-J. Li for using their LIBSVM.