Learning Cinematography from Annotated Films

David R. Winer DRWINER@CS.UTAH.EDU
School of Computing, The University of Utah, Salt Lake City, UT

1. Introduction

In this machine learning classification project, I use an annotated corpus of Western (genre) film scenes (Winer et al., 2017) that show a duel[1]. Because they are all duels, I expect that similar events are conveyed using similar cinematography. The features are *observations* associated with characters taking actions in a duel, and the labels are categories associated with 1) shot composition and 2) shot scale. After defining the data available and the instance space, I discuss challenges, introduce frequencies, and provide benchmarks, and provide results using bagging and boosting.

2. Observation Data

Observations are temporally bounded by at most the length of the shot they appear in. There is an observation for each entity that appears as an argument of an action in a shot and appears in the composition of the shot. A single observation is a tuple

$$obs = \langle e, A, start, fin, \mathbf{A}_e^{SH}, \mathbf{A}_e^{-SH}, \mathbf{A}^{SH}, \mathbf{A}^{-SH}, scene, d_s, f_s, n_s, t_s, scale, zeta, pos \rangle$$

featuring

- an entity e,
- an action type A (one of 37: $\mathbf{A}^{37}$) whose arguments (parameters) include e,
- $start \in \{0, 1\}$, a Boolean tag denoting if A is observed to start in the shot,
- $fin \in \{0, 1\}$, a Boolean tag denoting if A is observed to finish in the shot,
- $\mathbf{A}_e^{SH} \subseteq \mathbf{A}^{37}$, a vector representing actions observed in the same shot performed by e,
- $\mathbf{A}_e^{-SH} \subseteq \mathbf{A}^{37}$, a vector representing actions observed in preceding shot performed by e,
- $\mathbf{A}^{SH}$ and $\mathbf{A}^{-SH}$, vectors representing actions observed in the same and preceding shots, respectively, performed by entities other than e,
- $scene$, the name of the scene that the observation appears in,
- $d_s \in \mathbb{R}$, a real value duration of the shot (sec),
- $f_s \in [0, 1]$, the starting time (sec) of the shot divided by the total duration of the scene,
- $n_s \in \mathbb{Z}$, a non-zero shot number in the scene,
- $t_s \in \mathbb{Z}$, the total number of shots in the scene,

[1] A duel is where two or more gunmen have an escalating confrontation and face-off in a showdown.

- a scale value in $\{$cu, waist, figure, wide$\}$,
- $zeta \in \{0, 1\}$, a Boolean tag indicating foreground with 0 or background with 1, and
- $pos \in \{$left, centleft, center, full, centright, right$\}$, e's position in the first frame of shot n_s.

Data Cleaning In total, there are 30 scenes with 5709 observations in 1428 camera shots and about 140 action types. Observations where the position of the entity changes are removed for simplicity. Observation labels are binned (or removed) into one of the chosen classes. After binning and pruning, there are 3030 observations. I also eliminate observations whose action, A, is not in $\mathbf{A}^{37}$ (actions with at minimum 15 appearances), yielding 2716 observations.

Hypothesis Space The size of the hypothesis space for *shot scale* is $|H_{scale}| = 4^{37 \times \mathcal{C}(37)^4 \times 4 \times 3 \times 3}$, where the first 37 is for one of 37 action types, $\mathcal{C}(37)^4$ where $\mathcal{C}(n) = \sum_{k=0}^{n} \binom{n}{k}$ is for all permutations of the 37 action types for $\mathbf{A}_e^{SH}$, $\mathbf{A}_e^{-SH}$, $\mathbf{A}^{SH}$, $\mathbf{A}^{-SH}$, 4 is for start and finish Bool tags, and the final 3's are for shot lengths binned into $\{$short, medium, long$\}$ and for into-scene times binned into $\{$early, middle, end$\}$. The hypothesis space $|H_{pos}|$ has the same exponent as H_{scale} with base 6.

Train/Test Split I've randomly arranged the observations and saved 416 for a final test, leaving 2300 observations for training that are split into 5 folds (460 each) for cross validation.

Frequencies *Position Labels*: f(center) $= 1251$, f(right) $= 401$, f(left) $= 380$, f(centright) $= 326$, f(cent-left) $= 265$, f(full) $= 93$. *Scale Labels*: f(cu) $= 1125$, f(figure) $= 341$, f(waist) $= 446$, f(wide) $= 904$. Action frequencies are provided in Table 1. No label appears in more than 50% of observations.

3. Benchmarks

Naive Bayes I implemented the Naive Bayes algorithm to make a multi-class prediction. I used 5-fold cross validation with γ values $10, 6, 4, 2, 1$, results shown in Table 7. Using $\gamma = 10$, I ran on the whole training and on test set. On the scale training, acc=0.558, and on scale test, acc = .481. On xpos training, $acc = .499$ and on test, $acc = .454$.

Off-the-shelf Multiclass Logistic Regression I used an off-the-shelf L2-regularized Logistic Regression classifier (LIBLINEAR) (Fan et al., 2008) as a point of comparison. This test should therefore not count towards my effort on the project. I also used it as a point of comparison later in the paper on bagged forests. On scale, accuracy = 58.4135 (243/416), and on xpos, accuracy = 52.4038 (218/416). This seems fairly reasonable given the Naive Bayes output.

3.1 Decision Trees

My analysis of decision trees, created using the ID3 algorithm, are split into two parts: binary and multi-class. I modified the ID3 implementation from homework assignments to give multi-class

Table 1. Multi-class ID3s on Full Training Set. Depth is 27 for scale and 12 for xpos.

scale	precision	recall	f-score	xpos	precision	recall	f-score
cu	0.816	0.94	0.874	left,	0.807,	0.493	0.613
waist	0.966,	0.757,	0.849	cent-left,	0.742,	0.330,	0.457
figure	0.996,	0.867,	0.927	cent,	0.6251,	0.970,	0.76
wide	0.902,	0.866,	0.884	cent-right,	0.879,	0.289,	0.436
				right,	0.785,	0.321,	0.456
total acc:	0.89			total acc:	0.665		

Table 2. Train vs Test on Binary ID3. Depth is 27 for scale and 12 for xpos.

scale	training set	test	xpos	training set	test
cu	0.896	0.772	left	0.872	0.865
waist	0.857	0.856	cent-left	0.906	0.918
figure	0.880	0.875	center	0.819	0.673
wide	0.870	0.800	cent-right	0.881	0.880
			right	0.864	0.837

output, where entropy is modified to take sum of all label proportions instead of just positive and negative instances (i.e. -1 * sum(plogp(examples, value, total) for value in classes)).

Cross Validation For the multi-class ID3, I ran cross validation with depths 3 through 57, skipping every 3, and found that for scale, depth doesn't improve results after 27 with acc = .602, whereas for xpos, ideal depth is 12 with acc = 0.514. The results are in Table 8. For the binary ID3, I ran cross validation with depths 3 through 28, skipping every 5. The results are in Table 9.

Training Error First, I ran the best performing decision tree depths on the training and found that, for multi-class, scale performed at 0.89 accuracy and xpos performed at about 0.665 accuracy. I recorded precision and recall for each label (Table 1). The culprit for the poor performance on xpos (in multi-class) is low recall - the classifier seems to be biased against picking labels for the left and right, and prioritizes labels for the center. This led to precision-recall tradeoff reflected in lower precision for the center label. For binary decision trees, I calculated the best performing depth trees on training as well. The results are promising (Table 2) for predicting the binary presence/absence of individual labels.

Test Multi-class ID3 test results are found in Table 3, showing precision, recall, and f-score (harmonic mean) for each label. The multi-class results are impressive, superior to both the off-the-shelf multi-class logistic regression and multi-class Naive Bayes results. Binary ID3 test results are in Table 2. These show that binary ID3s perform well as weak classifiers.

Table 3. Test on Multiclass ID3. Depth is 27 for scale and 12 for xpos.

scale	precision	recall	fscore	xpos	precision	recall	fscore
cu	0.653	0.769	0.707	left	0.464	0.224	0.302
waist	0.707	0.577	0.636	cent-left	0.500	0.289	0.367
figure	0.800	0.667	0.727	center	0.554	0.925	0.693
wide	0.711	0.664	0.686	cent-right	0.857	0.120	0.211
				right	0.593	0.225	0.327
total acc:	0.692			total acc:	0.553		

4. Evaluation

Given the high performance using decision trees in the baseline performance test, I decided to see if I could increase performance in multi-class setting by building on the use of these trees. First, I implemented multi-class Adaboost by first implementing the binary case and then expanding based on details in Hastie et al. (2009). This algorithm uses binary decision stumps as input. Then, I created bagged forests of multi-class trees for varying depths and compare the bagged forest ensemble average with the multi-class logistic regression (which I introduced in baseline) over predictions as features (like in the homework, but for multi-class predictions).

4.1 Adaboost

First, I created binary decision stumps and used these stumps to test performance using Adaboost for multi-class output. I thought that since the binary decision trees were pretty successful, that a combination would be successful in a multi-class setting. I didn't test using multi-class decision trees because I was concerned that the multi-class trees needed greater depth to be successful, and these stumps are splitting on 1 to 4 features only, and that all labels may not be well represented by the output for shallow mullti-class trees.

First, I initialize weights to $1/m$ where m is the number of examples. Then, I calculate error for each classifier. Each classifier is a decision tree with either depth 1, 2, 3, or 4 where trees with depth d are given a distinct d number of features to build the stump, thereby distributing features to different trees (i.e. decision stumps receive 1 feature to split on, decision trees of depth 2 receive 2 features, etc.). Then, I pick the classifier with the smallest error, calculated as the sum of the weights for the examples that classifier picks wrong for. Then, I calculate the voting power α as $log(\frac{1-\epsilon}{\epsilon}) + log(k-1)$ for k classes. Then I check if we've finished by either hitting n rounds or if the total classifier is completely accurate. At each round, $H(x_i) = \sum_{t=0}^{c} \alpha_t * h_t(x_i)$ after $c-1$ rounds. If we are not finished, I update weights and rescale them so that the sum of correctly classified examples in this round sum to 1/2 and same among incorrectly classified examples. The weight update before rescaling is $w_{t+1} = w_t/2 + 1/(1-\epsilon)$ if correct and $w_{t+1} = w_t/2 + 1/\epsilon$ if

Table 4. Bagged Forests Ensemble Vote Train and Test

forest depth	train-scale	test-scale	train-xpos	test-xpos
5	0.463043	0.454327	0.497826	0.478365
8	0.501304	0.473558	0.497826	0.478365
11	0.546522	0.490385	0.511304	0.490385
14	0.56913	0.492788	0.518696	0.490385

Table 5. Multi-class logistic regression over forests of varying depth

depth	scale	xpos
5	0.709	0.579
8	0.695	0.526
11	0.707	0.543
14	0.712	0.57

incorrect, where ϵ is the error for the classifier that was chosen in the previous round. Then, I repeat by finding the next classifier that minimizes error over weights.

I did not receive good results using Adaboost with binary ID3s. Since the binary ID3s performed well individually, it's mysterious why they didn't work correctly. Given more time, I would run more tests to see why this is, including running with multi-class ID3s. The results were not better than chance (scale = .41 and xpos = .14). I also implemented a binary Adaboost and found comparable results to deep binary ID3s I reported earlier. Thus, I decided to abandon this path and focus instead on bagged multi-class trees.

4.2 Bagged Forests

I created multi-class ID3 forests for varying levels of depth and tested 2 ideas: *first*, I tested a voting ensemble and *second*, I tested the off-the-shelf multi-class logistic regression classifier. I created forests of size 1000 by randomly extracting 100 samples with replacement, as we had in our homework assignment. I created forests for depths $5, 8, 11, 14$ in order to test whether increasing depth would impact results. The results for the voting ensemble are in Table 4.

Last, I created revised training and test sets for each decision tree output in a given forest (for specific level of depth). Each label is assigned a new feature id and its output is always 1, to match the needed binary criteria. A feature id is created for each possible lable of each possible decision tree (i.e. the first 4 feature ids are the 4 labels for the first decision tree in the forest). The results for test on each depth are in Table 5, and reveal the best performance found among the different methods. There seems to be little relationship between the depth of the forest and performance, so shallow forests are likely sufficient.

5. Discussion

Overall, the scale test performed better than the xpos test. Training error was also lower for the xpos. In general, I bet that the type of action would have little influence on the x position of actions or actors on screen. In hindsight, I would have made two classes: a lateral class and a center class, where lateral covers all non-center x positions. On the other hand, I was pleasantly surprised to observe that scale was a predictable feature, getting over 70 percent accuracy on the final test. I was surprised at the successful performance of multi-class decision trees, especially for the logistic regression over the bagged multi-class trees. However, I am disappointed that I could not get Adaboost to behave as predicted in the multi-class output setting. This is likely because of some overlooked calculation, but more testing is needed. It took time to implement and I spent several hours reading and watching videos explaining the procedure.

Given more time, I would have added two additional features: the number of actions observed in the shot, and the number of actions observed in the previous shots. This might help because camera shots which are loaded with actions in a short amount of time are naturally going to be shot differently than long shots with a small number of actions. In general, these additional features would help characterize the timing of appearances of actions to greater specificity.

I would also have liked to test whether adding the xpos as a feature would have improved performance for the scale prediction, and vice versa. However, the ultimate goal here is a multi-label task - (i.e.) to predict both xpos and scale. The idea is that all of the features of the cinematography can be predicted as part of a sequence and used to film observations in Western duel scenes. Although some details about the sequence are characterized by the features included here (e.g. prior shot, duration into scene), it may be fruitful to extend the search backwards (i.e. k-gram for $k > 1$). Also, it would make sense to use the labels of surrounding features (i.e. what shot was used before and after this one, as part of the holistic editing sequence).

References

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., & Lin, C.-J. (2008). Liblinear: A library for large linear classification. *Journal of machine learning research*, *9*, 1871–1874.

Hastie, T., Rosset, S., Zhu, J., & Zou, H. (2009). Multi-class adaboost. *Statistics and its Interface*, *2*, 349–360.

Winer, D. R., Magliano, J. P., Clinton, J. A., Osterby, A., Ackerman, T., & Young, R. M. (2017). A specialized corpus for film understanding. *The AIIDE-17 Workshop on Intelligent Narrative Technologies (INT)*. AAAI.

Table 6. Actions by appearance in observations

stare-at	414
look-at	258
walk	200
provoke	194
fire-gun	174
look-from-to	174
arrive	122
draw-gun	102
square-off	90
aim-gun	86
de-escalate	71
assent	70
adjust-clothing	68
fall	62
leave	58
get-shot	53
taunt	45
give	40
rideto	39
identify	37
open	31
wince	28
die	27
face-from-to	26
stand-up	25
holster-gun	24
pick-up	23
side-step	22
face-at	21
drop	20
ask-for	17
raise-gun	17
drop-gun	17
cock-gun	16
lower-gun	15
chitchat	15
load-gun	15

Table 7. Cross Validation: Naive Bayes

γ	scale acc	xpos acc
10	0.522	0.485
6	0.524	0.463
4	0.525	0.452
2	0.516	0.442
1	0.519	0.430

Table 8. Cross Validation ID3-Multi-Class

depth	scale	xpos
3	0.449022	0.499239
6	0.483587	0.505
9	0.509891	0.512283
12	0.533696	0.514239
15	0.555217	0.500978
18	0.576196	0.495652
21	0.483152	0.483152
24	0.596413	0.48
27	0.601739	0.476522
30	0.603152	0.473261
33	0.604022	0.469457
36	0.605761	0.46663
39	0.606522	0.465543
42	0.606087	0.464783
45	0.60587	0.464348
48	0.605978	0.46413
51	0.605978	0.46413
54	0.606087	0.464022
57	0.606087	0.464239

Table 9. Cross Validation ID3 Binary per Label

	cu	waist	figure	wide	left	cent-left	cent	right	cent-right
3	0.593696	0.835543	0.870217	0.721957	0.86	0.897826	0.533913	0.877065	0.854022
8	0.635652	0.843696	0.873478	0.747283	0.849783	0.889783	0.562065	0.865761	0.84663
13	0.668478	0.845978	0.871413	0.763913	0.844348	0.881848	0.590761	0.851739	0.82663
18	0.692609	0.848261	0.872717	0.777283	0.832174	0.874022	0.598804	0.842065	0.816957
23	0.704891	0.846304	0.871087	0.77337	0.823587	0.866957	0.610652	0.835	0.807283
28	0.709783	0.845652	0.872391	0.772065	0.816087	0.862283	0.614565	0.831522	0.802717