Commercial Detection in MythTV using Bayesian and Neural Networks

Paul Turpin pturpin@gmail.com
University of Oklahoma, Norman, OK 73019 USA

Evan Stenmark stenmark@gmail.com
University of Oklahoma, Norman, OK 73019 USA

Abstract

Flagging of commercials in TV programming in addition to skipping these flagged sections can save the viewer considerable amounts of time and make the television viewing experience more pleasurable. The open-source Linux media software suite, MythTV, provides a framework for recording TV shows and detecting commercials. Using MythTV to produce commercial detection characteristics such as blank frame detection and network logo detection, we used these characteristics in our own machine learning agents to provide a better commercial detector than MythTV's own heuristic detector. The Bayesian network obtained an average accuracy of 95.9% and the neural network agent 94.4% both of which surpassed MythTV's average accuracy of 93.5%.

1. Introduction

Commercials have been a necessary evil when watching television programming for decades. With VCRs, it is possible to record and fast forward through commercials, but the process is slow and tedious. With the advent of the Digital video recorder (DVR), TV programs are encoded in MPEG2 format and saved digitally to a hard drive. Skipping through advertisements in a recorded video file is much simpler and faster than a VCR since interval time steps can be instantly skipped with the press of a button.

Detecting advertisement blocks in television shows is deployed in several commercial DVR set top boxes such as ReplayTV. There is non-free software for PCs

that have commercial detection such as BeyondTV and SageTV. Additionally free software DVR programs exist such as GB-PVR, MediaPortal, and MythTV. MythTV is an open source media center software suite for Linux that includes a DVR and commercial detection.

Commercials can be detected using indicators from television shows such as scene changes and detection of network logos. Flagged portions of a video file can then be instantly (and automatically) skipped while viewing the recording. MythTV provides framework for extracting these characteristics. This process is beyond the scope of this paper, but a good overview can be found in (Duan et al., 2006).

Our approach uses MythTV and its experimental detector to process the video file. Commercial detection indicators, generated by MythTV, and other characteristics of the video are then passed into two different machine learning networks: a Bayesian network and a neural network (NN). The networks output whether the given section of a video is a commercial or part of the show.

2. Problem Definition

Each of the commercial detection agents will be acting upon previously recorded TV shows. Each of the shows has been recorded from digital over the air (OTA) broadcasts. Table 1 lists shows the shows and some of of the characteristics—these are the recordings that have been analyzed.

In a recorded television show with a couple of minutes of padding at the beginning and end, 37% on average will be unwanted portions of the show. The unwanted portions are generally commercials, but they can also be the end of a previous show at the beginning of the recording or the start of the next show at the end of the recording. The job of each agent will be to analyze each recording and "mark" which portions of the show are unwanted. The agents will also be given recorded shows where the unwanted portions have been marked by a person. From these show the agents will learn what portions are unwanted so that they can mark future recordings.

A considerable amount of time stands to be gained if commercials can be automatically detected and skipped. However, this gain would be negated if intervention must constantly be taken to go back and view wanted portions of the show that were skipped when they should not have been

Show	Show Length (minutes)	Type	Original Broadcast Type	Category
Friends	30	Live Action	Analog	Comedy
Smallville	60	Live Action	Digital	Action
Heroes	60	Live Action	Digital	Drama
King of The H	30	Animated	Analog	Comedy
The Office	30	Live Action	Digital	Comedy
Bionic Woma	60	Live Action	Digital	Action
Reaper	60	Live Action	Digital	Comedy
24	60	Live Action	Analog	Action
Survivor: Chi	60	Live Action	Analog	Reality
Journeyman	60	Live Action	Digital	Drama

Table 1: Characteristics of the shows to be analyzed

o

r to force skipping of unwanted portions that were not correctly marked.

3. Agent Implementation

Four different methods will be used to analyze the recordings. The first is a random agent and the second a heuristic based method. The final two methods are machine learning based agents one of which will use a Bayesian network approach while the other will use Neural Networks.

3.1 Random Agent

The random agent uses a Markovian chain using the state space shown in Figure 1. The initial state is chosen at random with an equal probability of being either "commercial" or "show". From this point on the agent classifies blocks of 30 frames as either commercial or show based upon its current state. In between each classification step the agent will transition to the other state with a probability of 0.1. This continues until the entire recording has been classified.

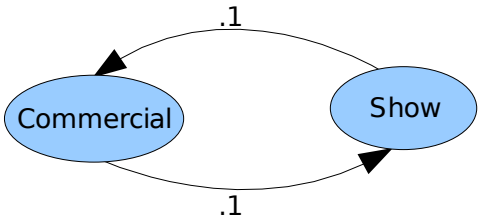

Figure 1: State space and transition probabilities for random agent.

3.2 Heuristic Method (MythTV)

The heuristic method used in comparison is the Experimental Commercial Detector as currently implemented in MythTV. It breaks the recording into blocks using black frames found at multiples of common commercial lengths and uses a combination logo presence, aspect ratio, black frame occurrences within the block and block length to classify the blocks. More details can be found at the authors website ("Commercial Detection").

3.3 Bayesian Network Agent

TV shows contain numerous blank frames. These are frames that are frames that are monochromatic; generally all black, but they can be all white or other colors. Usually the switch between commercial and content is delimited by either a single blank frame or a series of blank frames. The Bayesian agent works by breaking each recording into non-overlapping blocks based on these frames. Each block is then categorized as commercial or content using a conditionally weighted log likelihood algorithm based off of the Bayesian network shown in Figure 2.

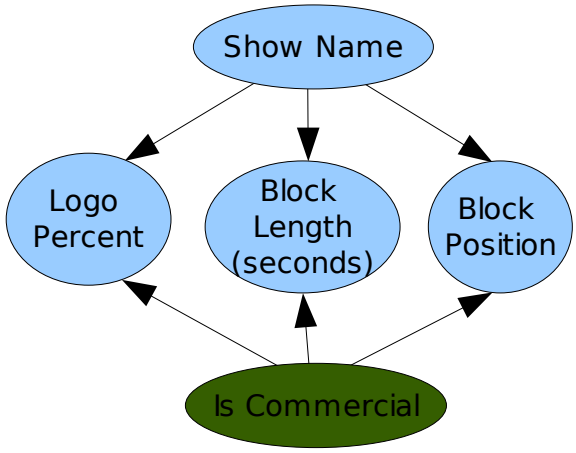

Figure 2: The Bayesian network used to score each block. The Blue nodes represent observable values.

The three attributes logo percent, block length and block position are continuous values and are converted to discrete value by means of a "bucket" method. Boundaries are defined and the values are placed into buckets based upon which boundaries the value falls between.

Equation 1 show the calculation of a likelihood value, L'. When L is greater than one it is more "likely" that the block is a commercial, while when it is less than one it becomes more likely that the block is show content. Equation 2 shows the result of taking the natural logarithm of both sides of the equation to get the log likelihood. In this version values greater than zero indicate that it is more likely that a block is a commercial while negative values indicate the opposite. We will call this L'.

$$L=\frac{p(c|g,t,l,p)}{p(!c|g,t,l,p)}=\frac{\frac{p(c,g,t,l,p)}{p(g,t,l,p)}}{\frac{p(!c,g,t,l,p)}{p(g,t,l,p)}}=\frac{p(c,g,t,l,p)}{p(!c,g,t,l,p)}$$
$$L=\frac{p(c)p(g|c,t)p(l|c,t)p(p|c,t)}{p(!c)p(g|!c,t)p(l|!c,t)p(p|!c,t)}$$

Equation 1: The likelihood (L) is based off the ratio of the probability of a commercial (c) to the probability of not a commercial (!c) based off logo presence (g), show title (t), block length (l) and block position(p)

$$\begin{aligned}\ln(L)&=\ln\left(\frac{p(c)}{p(!c)}\right)+\ln\left(\frac{p(g|c,t)}{p(g|!c,t)}\right)\\&\quad+\ln\left(\frac{p(l|c,t)}{p(l|!c,t)}\right)+\ln\left(\frac{p(p|c,t)}{p(p|!c,t)}\right)\\&=L'\end{aligned}$$

Equation 2: Log likelihood, L'

L' can also be expressed as the summation of scoring functions on an attribute a with value x,

$$L'=\ln\left(\frac{p(c)}{p(!c)}\right)+\sum_{a\in g,l,p}S_a(x) \quad ,$$

where

$$S_a(x)=\ln\left(\frac{p(a=x|c,t)}{p(a=x|!c,t)}\right) \quad .$$

Our experimentation has shown that some attributes may be a strong indicator that a block is commercial or content while it might not be such a strong indicator for the opposite. To harness this indication without introducing too many false positives we transform $S_a(x)$ into a conditionally weighted function $S'_a(x)$,

$$S'_a(x)=w_a \times S_a(x) \quad ,$$

where

$$w_a = \begin{cases} w_{a1} \text{ if } S_a(x)>0 \\ w_{a2} \text{ if } S_a(x)\le 0 \end{cases} \quad .$$

Good values for W_{ai} are established by an analysis of a few sample shows. The weight values used by the Bayesian agent are shown in table 2. This leads us to the final scoring equation, Score, used by the agent.

$$Score=\ln\left(\frac{p(c)}{p(!c)}\right)+\sum_{a\in g,l,p} S'_a(x)$$

If Score is greater than zero the block is flagged as a commercial, otherwise it is considered content.

Weight Values		
Attribute	$S_a(x)>0$	$S_a(x)\le 0$
Logo	2.25	2.00
Length	3.25	2.25
Position	2.50	0.75

Table 2: Weight values used by the Bayesian agent.

3.4 Neural Network Algorithm

The NN commercial detecting agent takes on a different approach to the problem than the Bayesian network. Rather than separating the video into segments between scene changes, the NN takes in each frame and its accompanying attributes and outputs whether each individual frame is a commercial or not.

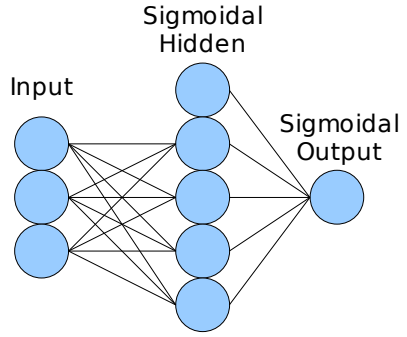

Figure 3: The simplified structure of the neural network. The full structure includes 10 input nodes including bias, 20 sigmoidal hidden layer nodes, and one sigmoidal output node.

The NN is similar to that given as an example by Bishop (Bishop, 2006). It is a 3 layer feed forward NN with backward propagation. The input layer is weighted and summed into the hidden layer which takes the logistical sigmoid of the sum.

$$a_j=\sum_i x_i \cdot w_{ji}$$
$$z_j=\sigma(a_j)$$

Since the output needs to be binary, '1' for commercial and '0' for show. The output is also a logistical sigmoid function of the sum of the weighted hidden layer nodes.

$$a_k=\sum_j z_j \cdot w_{kj}$$
$$y_k=\sigma(a_k)$$

Backpropagation is used in the NN to adjust the weights during training. To determine the change in the weights during each iteration, the error between the output, y, and the correct output, t, is found using the error function,

$$E_n=\frac{1}{2}\sum_k (y_k-t_k)^2 \quad ,$$

which is then minimized by taking a derivative, finding the delta for each weight,

$$\begin{aligned}\delta_k &= \frac{\partial E_n}{\partial a_k} \\ &= (\sigma(a_k)-t_k)\cdot(\sigma(a_k)\cdot(1-\sigma(a_k))) \\ &= (y_k-t_k)\cdot(y_k\cdot(1-y_k))\end{aligned} \quad ,$$

and updating the weight based on the delta and a small value parameter, eta,

$$w_{kj}=w_{kj}-\eta(\delta_k z_j) \quad .$$

The same is done for the hidden layer nodes.

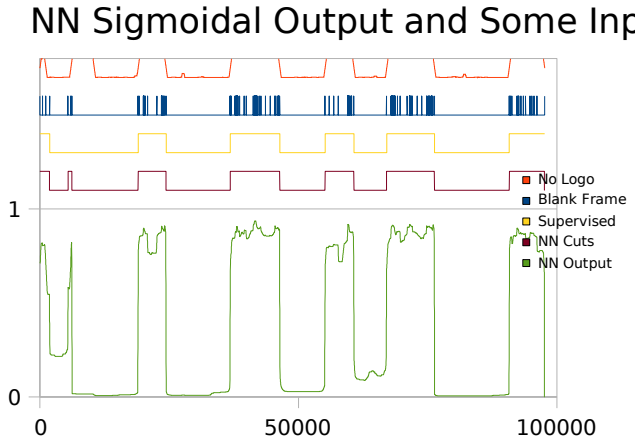

Figure 4: This shows the sigmoidal output of the NN (bottom, green). Also shown is whether or not a logo was present in the frame (1 – not present); whether or not the frame was blank (1 – blank); the commercial break points determined by a human (1 – commercial); and the commercial break points determined by the NN (1 – commercial).

$$\delta_j = \sigma(a_j) \cdot (1 - \sigma(a_j)) \cdot \sum_k w_{kj} \delta_k$$

$$w_{ji} = w_{ji} - \eta(\delta_j x_i)$$

The NN structure (shown in Figure 3) uses 10 inputs including the bias, 20 hidden layer nodes, and one output node. The inputs into the system are kept small (roughly between 0 and 2) so that any one input does not have a large variance compared to any others. Each frame of a recording is processed and the weights are updated for each frame. The number of frames could range from about 60000 for a half-hour long show to 120000 for an hour long show. Analyzing the frames of the show in order from start to finish can weigh the last part of the show more heavily than the rest. In order to alleviate this, the frames are analyzed in a randomized order.

In the Bayesian model, the distance between two blank frames is referred to as the block length. This block length is also used as an input to the NN. Each frame inside a block has that block's length as an input. Two related inputs were also used for each frame – the length of the previous block and the length of the next block.

The next input into the system is whether or not the frame being processed was determined by MythTV to be a blank frame. Blank frames typically indicate a scene change. This input alone is not particularly helpful because each frame is unaware whether or not adjacent frames are blank. Four "blank frame density" inputs were added later. Two densities sum the number of blank frames within 1000 frames of the current frame – one calculates to the left and another to the right. The other two blank frame densities make the same calculation but for within 5000 frames. Each density, Dn, is calculated for each frame, n, and whether or not the frame is blank, Bn,

$$D_n = \frac{\sum_{i=0}^{1000} B_n}{10},$$

and the result was divided by 10 to keep the values within the range of other inputs. Commercial sections generally have a lot of scene changes and blank frames accompanying them. When both the left and right densities are high, then typically the frame is a commercial, but if one density is high while the other is low, then that indicates a commercial/show boundary. Using the two different ranges (1000 and 5000) helps to eliminate spikes caused by sparse blank frames found in non-commercial spans.

Another input into the system is from MythTV's logo detector. Television networks often display their logo in a corner of the screen while playing a show and then remove it during commercials. This is typically a great indicator of commercial segments and the input is relied on somewhat heavily. Additionally, 500 frames are averaged to produce the value for each frame – this can be seen in Figure 4. This helps smooth the output where there are a few frames whose logo detection value is different from the majority

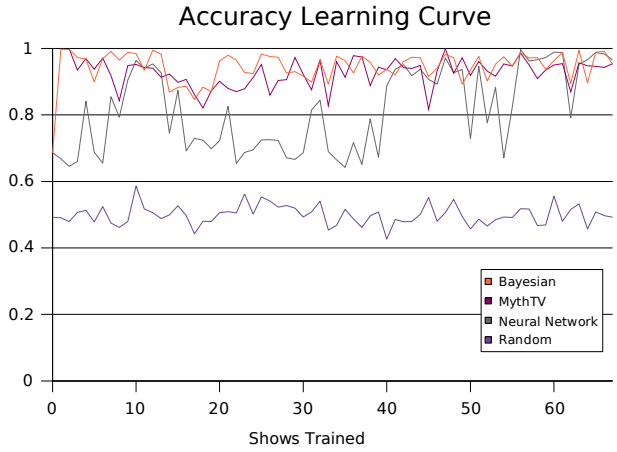

Figure 5: The ratio of the number of frames identified as show (non-commercial) to the total number of frames of show.

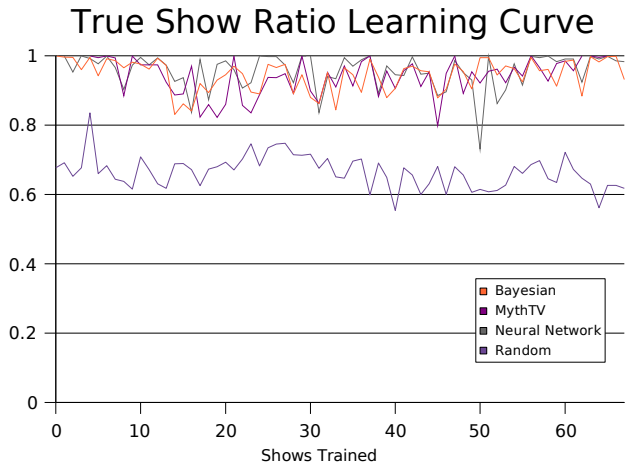

Figure 6: The ratio of the number of frames identified as show (non-commercial) to the total number of frames of show.

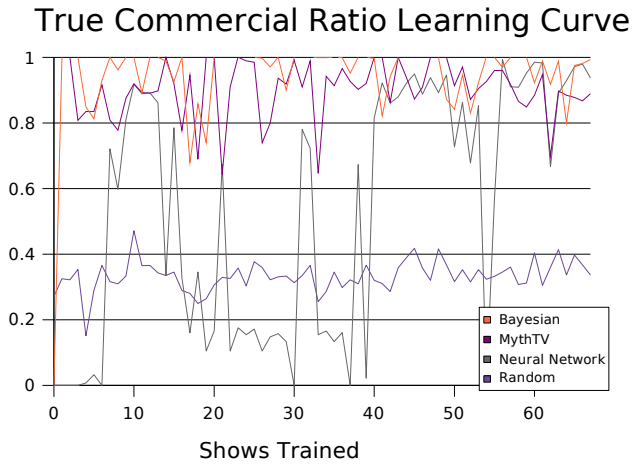

Figure 7: Ratio of correctly identified commercial frames. The erratic behavior of the NN agent is due to poor logo detection in some shows.

of the surrounding frames. Some of the recordings have had bad or incomplete logo matching from MythTV resulting in poor commercial detection. Because of this, another input was added that is a one if 15% or more of the frames have a logo detected (good logo detection) and a zero otherwise.

4. Experiments and Results

Two experiments were done to compare the different commercial detection approaches. The first, Train – Score – Train – Score (TSTS), utilized all of the shows during training and its initial set size was 68 shows. The second experiment used a training set and validation set out of 100 total shows. In both of these experiments, the MythTV detector did not train as it is not a learning method, instead it is shown for comparison. The Random agent is also compared in the experiments, but only to show a worst-case line that the learning agents should outperforming.

The performance results from the experiments are scored using three metrics: accuracy,

$$accuracy = \frac{correctly\ categoried\ frames}{total\ number\ of\ frames},$$

true commercial ratio (TCR),

$$TCR = \frac{correctly\ identified\ comm.\ frames}{total\ commercial\ frames},$$

and true show ratio (TSR),

$$TSR = \frac{correctly\ identified\ show\ frames}{total\ show\ frames}.$$

The true show ratio value is most important as it is better to misclassify a commercial as show than vice versa.

4.1 Train – Score – Train – Score ...

Each agent is run on a series of shows. With each show the agent first flags commercials.

After the agent has flagged a recording, if it is one of the machine learning based agents it then trains on the same recording using the human classifications. In total the initial data set consists of 68 recordings.

The results of this experiment is shown in Figures 5-7. These graphs illustrate how the NN needs more training to obtain a higher accuracy. The NN also was not as sophisticated during this experimental phase as it was during the second experiment described in section 4.2. Most notably during this experiment, the NN did not have the the input which told it whether the logo detection was valid or not. Figure 7 highlights the NN detector's inability to cope with invalid or incomplete logo detection from MythTV. The NN will flag no commercials in a recording with poor logo detection.

The Bayesian agent achieves a very high accuracy after

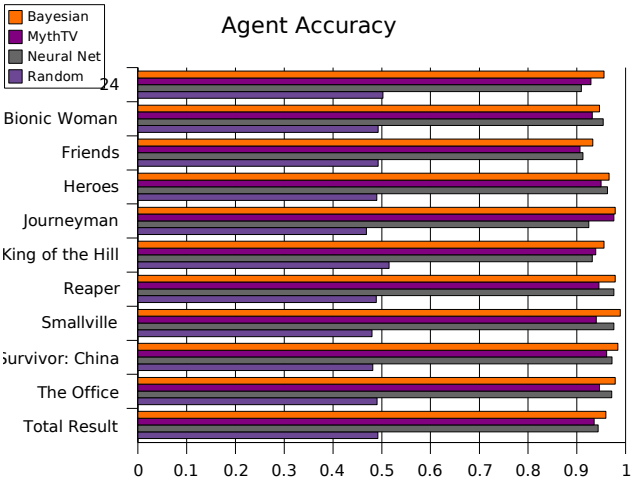

Figure 8: Accuracy of each agent by show after validated training. The Neural Net and Bayesian agents outperform MythTV.

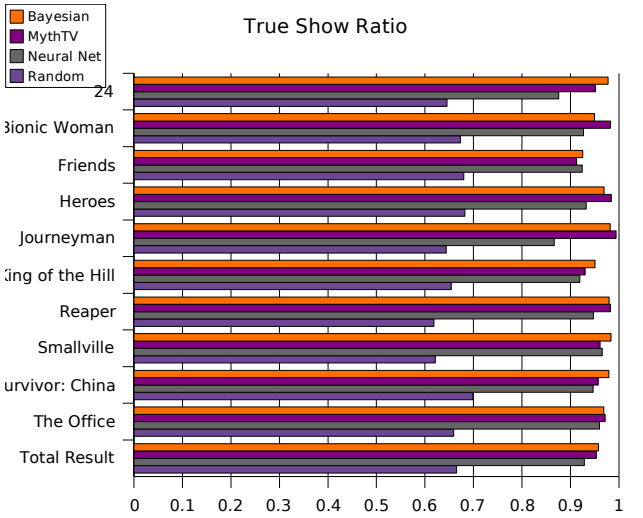

Figure 9: True show ratio broken down by the show name. This characteristic is more important than the True Commercial ratio as it is better to see a little bit more commercial than to miss part of the show.

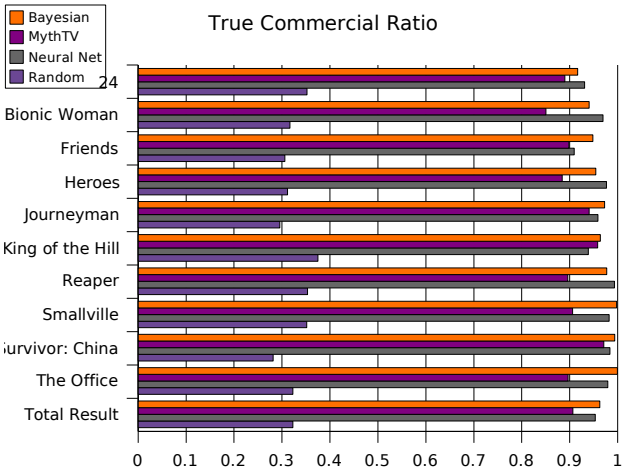

Figure 10: Average True commercial ratio by show.

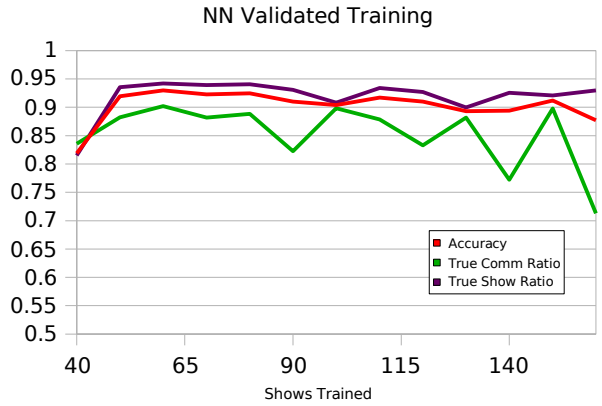

Figure 11: Validated training on the neural net. The performance starts getting eratic towards the end due to a high eta and over-training. The optimal agent parameters is found in the 60-80 shows trained range.

training on only a couple different shows. The MythTV detector continues to outperform the NN agent, but the Bayesian agent is clearly the top performer achieving the highest accuracy and solid TSR.

4.2 Validated Training

In order to reduce over-fitting while training a neural network or Bayesian network, a validation set is often used to measure the performance. When the performance of the learning agent begins to decrease on the validation set, then the agent has become over-trained on the training set.

The validation set was constructed by randomly selecting 2 recordings out of each of the 10 different show groups. The remaining 79 recordings were randomly ordered used as the training set. The agent would train on 20 of the randomly order shows in the training set and would then score its performance on each of the recordings in the validation set.

After a sufficient time in training, a curve can be constructed as shown in Figure 11. After finding the point where the validation set accuracy is the highest, those stored agent parameters (e.g. weights for the NN) can be used to find the performance of the agent on all of the shows. The performance of the agents (accuracy, TSR, and TCR) is shown in Figures 8-10. Each graph shows the average performance for all the recordings in one show group in addition to the overall average performance.

The Bayesian agent again surpasses both the MythTV commercial detector as well as the NN agent in all three metrics. The NN agent has an average performance greater than the MythTV agent in the categories of accuracy and TCR, but not in the true show ratio. This signifies that the MythTV detector's quality of flagging may be slightly higher than the NN agent because having a higher true show ratio with a slightly lower accuracy can produce better results than higher accuracy and true commercial ratio. The Bayesian performs better than the MythTV agent in both experiments which may possibly be due to its simpler structure—fewer inputs and simpler inputs.

Accuracy Result Summary

	Bayesian	Neural Net	MythTV	Random
Average	0.959	0.944	0.935	0.492
Standard Deviation	0.034	0.046	0.043	0.040
Minimum	0.828	0.788	0.818	0.364
Maximum	1.000	1.000	1.000	0.576

Table 3: This table shows some statistics for the accuracy values given in Figure 8.

The average TSR graph, Figure 9, illustrates which shows are the hardest to accurately classify. Each of the shows which were originally analog, Friends, 24 and King of the Hill, have lower average TSRs. This is most likely due to the noise of these recordings introduced by passing through various lousy conversions. The logo detection on these shows is generally not as accurate either.

The validated training experiment yielded better performance in addition to more reliable results. Though the Bayesian agent never showed signs of significant over-training, the training and accuracy of the NN agent was certainly improved using this method.

5. Development Notes

In order to keep the results somewhat simplified several items were not included in the main body of the paper. This section is a brief synopsis of some of those items that may be of interests.

Originally it was hoped that closed captions indicator of commercial vs show. However no words from the closed captions ended up being good indicators of a commercial and in general ended up hurting the results more than helping them. Additionally any increase in complexity to the Bayesian network generally had the impact of degrading the results.

An additional agent that was created that is worthy of note is the combined agent. This agent attempted to combine the outputs of the Bayesian network, neural network and MythTV agents on a frame by frame basis using a linear regression method where the weights were incrementally learned. While this agent did perform well, its performance generally ended up between that of the neural network agent and the Bayesian network agent.

6. Future Work

One downside of the weights in the conditionally weighted log likelihood is that they are chosen by hand based off an analysis of the data. It could be interesting to apply some machine learning technique to the choice of these weights.

For the NN agent, it would be good to add a second, separate agent that analyzes the output of the NN agent. The second agent would smooth out the output of the NN agent in addition to removing spikes that may have occurred in the output of the NN. Most importantly, the second agent would provide a variable commercial threshold value (rather than the static one currently

implemented) based on the characteristics of the output graph.

While both methods explored were supervised learning methods, this is not the easiest method to use for an end user. It would be interesting to see if a reinforcement learning method which uses the skipping forward and backwards through the show as a basis for rewards is able to achieve a reasonable level of accuracy.

7. Conclusion

Bayesian network and Neural network based agents are able to accurately detect commercials within recorded TV programs with accuracies similar to those of the current MythTV heuristic based method. With our current preliminary results MythTV obtained an accuracy of 93.5% while the Bayesian agent obtained 95.9% and the Neural network agent 94.4%.

To go along with the spirit of open source software which was leveraged significantly in these studies the commercial detection agents developed will be submitted back to the MythTV project so that it may be incorporated into MythTV at the discretion of that project.

References

Duan, L., Wang, J., Zheng, Y., Jin, J., Lu, H., and Xu, C. (2006). Segmentation, Categorization, and Identification of Commercials from TV Streams Using Multimodal Analysis. *Proceedings of the 14th Annual ACM International Conference on Multimedia.* (pp. 201-210).

Sutton, R. S., & Barto, A. G. (1998). Reinforcement Learning: An introduction. MIT Press: Cambridge, MA.

Bishop, C. M., (2006). Pattern Recognition and Machine Learning. Springer: New York, NY. (pp. 245-246).

"Commercial Detection", http://www.tsaiberspace.net/projects/mythtv/commercials/, accessed on 12 Sept. 2007