CSE, IIT BOMBAY

BTP Report

Aditya Kumar Akash, 120050046
October 9, 2015

1 TAG VIDEO GAME

A Game where users could tag the video by selecting from a fixed set of tags. The user is given some score based on how much relevant the tag is to the video.
A good scoring function is need for following reasons -

1. Maintain user interest to give better and relevant tags
2. Not allow users to tame the game by some global consensus to give irrelevant tags

1.1 SCORING FUNCTION

Earlier scoring function of the game uses score given by IBM Watson and tf-idf scoring to generate the scores. Following is the scoring -

term frequency $tf = \frac{f_v^t}{c_v}$,
$t = tag, v = video$, f_v^t =frequency of tag t, c_v =distinct number of tags for video v

$idf = 1 + \log \frac{N}{N_t}$, N =Total Videos, $N_t = v$ videos where tag t is present

$tfIdf = tf * idf$, $max_v(tfIdf)$ =max $tfIdf$ for video v

weight $w_{watson} = \frac{1}{e^{c_0 * (videoTaggers - 1)}}$ given to watson score

$$score = s_{watson} * w_{watson} * idf * c_1 + (1 - w_{watson}) * \frac{tfIdf}{max_v(tfIdf)} * c_2 + c_3$$

1.2 Problems with Scoring and Improvement

Some inherent problems with old scoring -

1. $tf = \frac{f_v^t}{c_v}$, division by count of distinct tags not necessary since this gets canceled out in the final score calculation
2. Presence of $\frac{tfIdf}{max_v(tfIdf)}$ is just to get score in range (0, 1). Extra information stored which is not required
3. $\frac{tfIdf}{max_v(tfIdf)} = \frac{tf_1 * idf_1}{tf_2 * idf_2}$, subscript 2 corresponds to max value case
4. idf_2 can be reduced by tagging other videos with tag 2, which causes score increase for some junk tag 1 without any user specifying the tag 1. This is not desirable

Suggested improvements on similar ideas which deals with all the above problems is as follows -

1. $tf = \frac{f_v^t}{f_v}$, division by maximum frequent tag, Handles decrease in tf of junk tag in long run
2. idf_2 removed and normalization of idf by division by $1 + logN$
3. Above tf and normalization ensure scores between 0-1 along with property that junk tag scores decreasing with time

New scoring function -

$$tf = \frac{f_v^t}{max(f_v)},\ t = tag, v = video,\ f_v^t = \text{freq of } t$$

$$idf = 1 + \log\frac{N}{N_t},\ N = \text{Total Videos},\ N_t = v \text{ where } t \text{ is present}$$

$$tfIdf = tf * idf,\ Idf_{max} = 1 + \log N$$

$$\text{weight } w_{watson} = \frac{1}{e^{c_0 * (videoTaggers - 1)}} \text{ given to watson score}$$

$$score = s_{watson} * w_{watson} * \frac{idf}{Idf_{max}} * c_1 + (1 - w_{watson}) * \frac{tfIdf}{Idf_{max}} * c_2 + c_3$$

1.2.1 Experiments on Scoring Function

Old Scoring

After **S4** 9 more videos where tagged *Iron Man*, then **S5** onwards proceeded to re-tag this video.

Tag	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**S9**
Iron Man	80	100	100	100	-	-	100	100	-
Metal	0	10	-	-	40	90	-	-	100

Table 1.1: Tag Score for Video - Will you be iron man?

'-' denotes this tag was not given

New Scoring

After **S4** 9 more videos where tagged *Food*, then **S5** onwards proceeded to re-tag this video

Tag	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**S9**
Food	70	75	90	100	-	-	95	95	-
Shoe	0	0	-	-	20	60	-	-	75

Table 1.2: Tag Score for Video - Science of sweetness

'-' denotes this tag was not given

Following could be observed in the implementation of the new score -

1. The old scoring does not reflect clearly idf factor affecting the score
2. Old scoring allows unrelated tags have spiked increase in score
3. New scoring handles these cases well

2 Meta Learner

In the previous case we observe that there is no learning about users of video categories done in previous case. So we would like to design a meta learner which would make use of the data available in form of (video, tag set) pairs to learner video categorization. We try to look into the kind of problem at hand.

2.1 PROBLEM

We need a meta learner which would evolve with each video, tag set provided by the user. The problem is closely related to multi-label classification problem in unsupervised settings.
The final aim of learner would be to classify the video based on feature set into a class consisting of different labels or to reach a consensus based on labels given by users.

2.2 BOOSTING KIND OF META LEARNERS

The most popular kind of meta learners available are the learners using boosting. We need an online boosting kind of framework for our problems. So we considered online ADABOOST. Here a sequence of base models $h_1, h_2, ...h_M$ are trained using weighted training sets. In online framework each base model has training set consisting of K copies of each of original training examples where $P(K = k) = \frac{N}{k}\frac{1}{N}^k(1-\frac{1}{N})^{N-k}$ which is binomial. As $N \Rightarrow$ inf distribution of K becomes poisson.
The poisson parameter λ if miss-classified is increased otherwise decreased when presented to next base model.

In our problem we have following adaptations and concerns :

1. The boosting learners require training data with true labels. Need to generate initial training data to initialize the learner
2. We would treat each of our players as base learners and model them using the inputs we receive from them. Then use ADABOOST over them. The data available would be sparse so need some other online algorithm like LPBOOST.
3. The tag set we obtain from users are noisy and do not represent true label set.
4. We would use some static base learners like image classifiers, audio/metadata based classifiers and make adaboost meta learner using them. Then evolve this learner using the input from users.

Mainly because of sparsity of data and need to generate initial training examples we did not go ahead with boosting and looked for other directions.

2.3 CONSENSUS BASED LEARNERS

Consensus learning considers the problem of consolidating multiple supervised and unsupervised information sources by negotiating their predictions to form a superior classification solution.

Currently working on "Multi label consensus learning" where results from base models are combined to achieve maximum consensus among them. Here the problem is expressed as an optimization problem estimating the probability distribution for an instance belonging to given class and probability of seeing a label from another label.

1. For our problem case we would have to adapt the problem of handling online updates
2. We also have to adapt the given solution for sparse setting
3. We have to take into account the initial hierarchy structure among the classes rather and frame the problem accordingly.

Currently working on the above goals.