2013 IEEE/WIC/ACM International Conferences on Web Intelligence (WI) and Intelligent Agent Technology (IAT)

# Fully Automated Learning for Application-Specific Web Video Classification

Chetan Verma, Sujit Dey
Mobile Systems Design Lab, Dept. of Electrical and Computer Engineering
University of California San Diego
{cverma,sdey}@ucsd.edu

***Abstract*—Personalization applications such as content recommendations, product recommendations and advertisements, and social network related recommendations, can be quite beneficial for both service providers and users. Such applications need to understand user preferences in order to provide customized services. As user engagement with web videos has grown significantly, understanding user preferences based on videos viewed looks promising. The above requires ability to classify web videos into a set of categories appropriate for the personalization application. However, such categories may be substantially different from common categories like Sports, Music, Comedy, etc. used by video sharing websites, leading to lack of labeled training videos for such categories.**

**In this paper, we study the feasibility and effectiveness of a fully automated framework to obtain training videos to enable classification of web videos to any arbitrary set of categories, as desired by the personalization application. We investigate the desired properties in training data that can lead to high performance of the trained classification models. We then develop an approach to identify and score keywords based on their suitability to retrieve training videos, with the desired properties, for the specified set of categories. Experimental results on several sets of categories demonstrate the ability of the proposed approach to obtain effective training data, and hence achieve high video classification performance.**

## I. Introduction

Over the past few years, there has been a steady rise in the number and popularity of personalization applications available on the Internet. These include personalized advertisements, content recommendation systems, social network connection suggestions, and several others, that attempt to understand the preferences of users. Personalization applications have been traditionally based on learning user preferences through queried keywords and viewed articles. The last few years have also witnessed a tremendous increase in viewing and sharing of web videos (such as on YouTube), with significant increases in unique viewers, total streams viewed, number of streams per viewer, and the time per viewer [1]. Given their unique characteristics, web videos offer a tremendous potential for understanding user preferences.

User preferences can be inferred based on the types/categories of web videos seen. Such videos are generally organized at video sharing websites on the basis of labels that the video uploaders choose from among a set of common categories that are used by such websites. Examples of such common categories include Comedy, Music, People, Entertainment, Pets, Science, etc. On the other hand, the categories of interest to personalization applications may be arbitrary, and quite different from the above common categories. Consider a department store (such as Sears or Walmart) that might want to offer promotional coupons to buyers. Knowing whether a person (a buyer) has interest in product specific categories like fitness equipment, clothing items, or baby products would be of high interest to the department store, as compared to knowing whether he/she is interested in the common categories mentioned above. A movie recommendation system would like to learn if a viewer prefers action, horror, or comedy movies. Categorizing viewed videos and understanding user preferences in terms of the common categories used by video sharing websites might not be useful for different personalization applications. In addition to the above observation, it should be noted that different personalization applications are interested in understanding user preferences with respect to very different sets of categories, as shown by the above examples. It is clearly not sufficient to use a common set of categories for every personalization application, as the categories of interest for one application might be irrelevant and useless for another.

This calls for techniques to classify viewed web videos, and hence estimate user preferences, in terms of any arbitrary set of categories appropriate for a given personalization application. Various modes of information (such as audio, visual, textual and social network) can be employed to assist in the classification of web videos. Classifiers employed for this task have the inherent requirement of training videos labeled to the set of categories as desired by the personalization application. Since the set of categories suitable for a personalization application might be very different from the common categories used by video sharing websites, training videos for the required set of categories are often unavailable. Our work addresses this requirement of labeled training videos for an arbitrary set of categories, which are not necessarily the categories commonly associated with web videos. We propose a fully automated framework to obtain training videos with properties that can lead to high performance of trained classification models. To achieve the above, the proposed framework neither relies on labels associated with online videos, nor requires any manual labeling of videos. Instead, we develop an approach to identify and score keywords based on their suitability to retrieve high quality training videos for a specified set of categories.

978-1-4799-2902-3/13 $31.00 © 2013 IEEE
DOI 10.1109/WI-IAT.2013.44


Fig. 1: (a) Sample training videos for categories: {*Baby, Clothing*}. Circled video is wrongly placed in category $Baby$, and is hence a mislabeled video. (b) Variety of video topics belonging to category $Baby$

### A. Related Work

There has been a significant amount of work addressing the problem of video classification. Such work can be looked at on the basis of two dimensions - modalities used for classification, and approaches to obtain labeled training videos. While the focus of our work is on obtaining training videos, we first briefly describe video classification approaches in terms of modalities used, including our approach, and then discuss approaches to obtain training data, contrasting our approach from others.

A characteristic property of web videos is that they have rich information in several modes – audio, visual, textual, and social network being the most common ones. Methods such as [2], [3], [4], [5], [6] present multi-modal techniques for classification of web videos. Others such as [7], [8] classify videos using only the audio-visual information in the videos, while [9], [10] approach classification of web videos by treating them as text documents. A detailed survey on video classification is provided in [11]. In our work, we classify web videos on the basis of the contextual information surrounding them, such as the title, keywords, and description. This is because text-based classification approaches are computationally much less expensive than multimedia features-based classification approaches, and as shown in existing literature as well as in Section IV, offer good classification performance.

In terms of approaches to obtain labeled training videos, [9], [10], [5] obtain training videos that are labeled according to categories used by YouTube. Hence, such approaches cannot be used for classification of web videos to arbitrary set of categories, which is the focus of this paper. Approaches such as [7], [6], [2] utilize training videos that are labeled manually. Recently, techniques have been developed [3], [4] which expand the set of training videos starting from a set of manually labeled videos. With the help of social network structure of the video sharing website, co-watched videos, or text-based classifiers, [3], [4] increase the number of training videos in a semi-supervised fashion. However, manual labeling requires human experts to go through at least a part of the video, and come up with a label. The labeling process is prone to human errors and inconsistencies, and more critically, is not scalable to large sizes of training data, especially given the enormous scale of web videos [1]. Contrary to these approaches, we propose a framework that does not require any manual effort to obtain training videos, even for any arbitrary set of categories desired by a personalization application.

For multi-class, single-label classification of web videos, we first discuss the desired properties of training videos that can lead to high performance of trained classification models. This is done in Section II. Section III describes our proposed approach of identifying and scoring keywords to retrieve training videos with the desired properties. Section IV discusses the experimental set-up and presents performance and complexity results. Section V concludes.

## II. Desired Properties of Training Data

For a given classification model, a good training data would be one that has no mislabeled instances, and has high Intra-Category Diversity for each category. We discuss both factors in this section. The goodness of training data is reflected in terms of its performance on a large test set.

In the domain of video classification, mislabeled instances refers to videos that have the label of category $i$ as per the training data, but in actual, belong to the category $j(\neq i)$ as per an oracle. For instance, consider the set of categories {*Baby, Clothing, Fitness, Food*} that a retailer may be interested in, to enable personalized promotions of the above product categories. If certain approach for obtaining training videos includes a video on $Shawls$ or $Trousers$ (as shown in Fig. 1a) to the set of training videos for $Baby$, the video would be a mislabeled video since its true label would be $Clothing$, but it has the label of $Baby$ in training data. A true label of a video is defined as the label that an oracle would assign to the video. [12], [13] discuss techniques to identify (and eliminate) mislabeled instances from training data, for classification tasks. The performance of classification models is shown to have increased considerably after identifying (or eliminating) mislabeled videos, thus supporting that less mislabeled instances is desired in training data.

By Intra-Category Diversity of training videos T($i$) of category $i$, we refer to the extent to which T($i$) encompasses the essence of category $i$. Let us denote Intra-Category Diversity

of T($i$) as $div(T(i))$. In order to first intuitively motivate why high $div(T(i))$ is desired, consider the same set of categories {*Baby, Clothing, Fitness, Food*}. Fig. 1b shows some of the various topics of videos that one would associate with the category $Baby$. A set of training videos T1, having videos on *Funny kids* only has less Intra-Category Diversity than a set T2 of same cardinality as T1 but having videos on *Funny kids, Newborn care, Babysitting*, and *Stroller reviews*. A classifier trained over T2 is expected to have higher likelihood of categorizing correctly a test video $v$ belonging to category Baby as compared to a classifier trained over T1. For instance, if a user watches a video related to review of popular strollers for infants, the model trained on T2 will find it more similar to the training videos on $Baby$ than the model trained on T1 will, and hence will have higher likelihood of categorizing it correctly.

As discussed later in Section III, one of the shortcomings of approaches that obtain training videos without manual labeling, is low Intra-Category Diversity. In such scenario, the training data of a category is skewed towards certain dominant themes within the category, and encompasses only limited topics of videos within the category. Improved techniques are hence required to obtain training videos with high Intra-Category Diversity. While it is understandable what $div(T(i))$ means, calculation of $div(T(i))$ requires knowledge of a) different topics within category $i$, and b) the extent to which these topics are covered by the training videos of category $i$.

For an arbitrary set of categories, it is extremely difficult to obtain (a) and (b) above without the help of an oracle. The Intra-Category Diversity, $div(T(i))$ of T($i$) can be estimated on the basis of the diversity (or variation) within the set T($i$). There exist several ways to estimate the diversity or variation within a set T($i$). Assuming a certain distance measure between instances (web videos, in our case) in a set, the diversity can be measured by the average pair-wise distance between the instances. The time-complexity of such a measure, however, varies as $O(N^2)$ where $N$ is number of training videos in category $i$. Number of intrinsic dimensions of T($i$) [14] is an alternative measure. While several measures for diversity (or variation) within a set exist, we choose to estimate $div(T(i))$ by the variance of T($i$), primarily because of its low time-complexity.

$$div(i) = \sqrt{\frac{\sum_{j=1}^{N} \mid v_j - \mu_i \mid^2}{N}}, \tag{1}$$

where $\mu_i = \frac{\sum_{j=1}^{N} v_j}{N}$; $v_j$ is a training video for category $i$, and $N$ is the cardinality of the set T($i$), i.e., $\{v_j \in T(i)\}_{j=1}^{N}$. In Section IV, we present numerical values for $div(T(i))$ for various categories, and experimentally verify that an increase in Intra-Category Diversity of training data translates to improved performance of the trained classifier.

## III. Our Approach

In this section, we describe the proposed framework, and our approach for identifying and selecting keywords to obtain training videos.

As discussed in Section I, obtaining training videos for arbitrary set of categories through manual effort is not scalable to large sizes of training data. Training videos could alternatively be obtained in an automated manner using a video search engine. This approach has also been briefly mentioned in [3], to obtain weakly labeled videos. Let RV($K$) represent the set of retrieved videos obtained by querying keyword $K$ in a video search engine, such as YouTube or Metacafe. The training videos of category $i$, i.e., T($i$) can then be obtained simply as RV($C_i$), where $C_i$ is the name of category $i$. Though simple, this technique has few shortcomings. It focuses only on occurrence of the name of category and not on its semantic meaning. For example RV('$Baby$') mainly retrieves videos on funny kids, and music videos or other popular videos containing the word 'Baby' in their title or tags. As a result, there are several mislabeled videos among training videos obtained by this approach. In Section IV, we show how this leads to poorer performance as more videos are retrieved just by the category name. At the same time, this approach does not cover the many of the semantic topics that we associate with the category of concern. For the category $Baby$, these include topics such as (in addition to funny kids,) Strollers and Bassinets reviews, Babysitting tutorials, Newborn care, Pregnancy, and several others (Fig. 1b). The Intra-Category Diversity by such an approach is hence quite low, leading to poor classifier performance. Through the proposed framework, we attempt to address the above shortcomings. We use the training videos obtained by querying name of category, i.e., T($i$)=RV($C_i$), to train baseline classifiers to compare with our proposed approach.

### A. Overview of Proposed Framework

In the proposed framework, we first collect several keywords that are related to the name of category ($C_i$). These comprise the **Candidate Keywords**. The Candidate Keywords (called candidates for brevity) can be obtained on the basis of correlation or co-occurrence with the name of the category from publicly available text documents (such as Wikipedia). Thesauri also provide a good source for semantically (i.e., in terms of meaning) similar keywords, and can be used to obtain candidates. The candidates can be queried in a video search engine, and their retrieved videos can be collected to obtain T($i$). However some candidates would be more useful than others, and some might be outright harmful, if used to retrieve training videos for category $i$. We discuss these precisely in the next section. On the basis of a proposed keyword selection algorithm, we select a subset of keywords from a set of candidates for category $i$. The Selected Retriever Keywords (or SRKs) thus selected are used to retrieve training videos. If $\{K_{i,1}, K_{i,2}, , K_{i,L}\}$ are the SRKs for category $i$, then

the training data for category $i$, i.e., T($i$), can be obtained as:

$$T(i) = \left[\bigcup_{j=1}^{L} RV(K_{i,j})\right] \cup RV(C_i). \quad (2)$$

If T($i$) were obtained as per (2) by selecting any arbitrary candidate keywords of category $i$ as SRKs, then T($i$) may not necessarily have the desired properties discussed in Section II, namely low mislabeled videos, and high Intra-Category Diversity. In the next section, we discuss how we can determine the suitability of candidates to retrieve training videos with the desired properties discussed in Section II.

### *B. Selection Procedure for Selected Retriever Keywords (SRKs)*

Before we discuss suitability of candidates, and the selection procedure for SRKs, we provide the following result to help in our discussions. Consider the set of categories $\{i\}$, where each category $i$ is a multivariate normal distribution with mean $\mu_i$. Assume that the set of videos across all categories (referred to as data) is whitened - i.e., has uncorrelated dimensions of variance unity. Then $\Sigma_i = I$, i.e., the covariance matrices for all categories reduce to an identity matrix. Under assumptions of equiprobable categories, it can then be shown that a video (represented as $v$) belongs to category $\hat{i}$ iff

$$\hat{i} = \arg\min_i \mid v - \mu_i \mid, \quad (3)$$

where $\mu_i$ is the mean of category $i$, as determined by an oracle. $\mid v - \mu_i \mid$ is the Euclidean distance between $v$ and $\mu_i$. Note that for this paper, $v$ is represented as a bag-of-words vector based on the contextual information surrounding $v$. On the basis of the above assumptions, and using (3), a keyword $K$ retrieves more videos having true label as category $i$ than videos having true label as category $j(\neq i)$, if

$$\sum_{v:v\in RV(K)} I(\arg\min_l \mid v - \mu_l \mid = i) \quad > \quad \sum_{v:v\in RV(K)} I(\arg\min_l \mid v - \mu_l \mid = j) \ \ \forall j \neq i. \quad (4)$$

Here, $I(.)$ is an indicator function that is 1 if its argument is true, and 0 if its argument is false. $\mu_l$ is the true mean of category $l$. In (4), the closest category for each video in RV($K$) is obtained, in terms of Euclidean distance. Checking (4) for a keyword $K$ thus has complexity $O(\mid RV(K) \mid .N_{Cat})$, where $N_{Cat}$ is the number of categories, and $\mid RV(K) \mid$ is the cardinality of the set RV($K$). In order to reduce the above high complexity, (4) can be approximated by obtaining the closest category of centroid of the set RV($K$). This reduces the complexity to $O(N_{Cat})$. For category $i$, we define **Valid Candidate Keywords** (called valid keywords for brevity) as those Candidate Keywords that retrieve more videos having true label of category $i$ than of any other category. Then for a candidate $K$, $K$ is a valid keyword if

$$\mid \mu_K - \mu_i \mid < \mid \mu_K - \mu_j \mid \forall j \neq i. \quad (5)$$

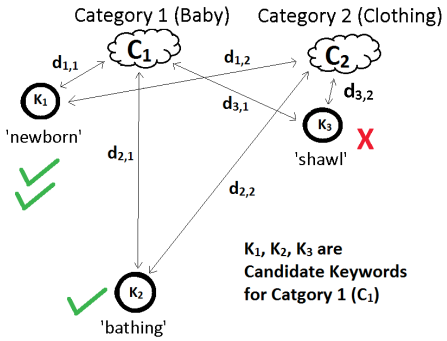


Fig. 2: Selection Criteria for Selected Retriever Keywords

Here $\mu_K = \frac{\sum v:v\in RV(K)}{|RV(K)|}$. The true mean $\mu_i$ for category $i$ can be approximated as centroid of RV($C_i$), i.e., of the set of videos retrieved by name of category $i$. Equation (5) is called the **Validity Filter**. Only valid keywords should be considered for being selected as SRK to ensure more number of training videos are added in T($i$) that have true label of category $i$ than videos that are mislabeled as category $i$.

Note that we have assumed that the data generating the above distributions is whitened. If the given data is not whitened, its dimensions can be rotated into space of principal components, and each dimension be divided by square root of variance in that dimension, in order to whiten the data. Also, if the assumptions of equiprobable multivariate normal distributions as categories do not hold, then exact analysis can be carried out using true distributions. For simplicity, we continue our discussion with the above assumptions.

Let us consider the scenario shown in Fig. 2. Keywords $K_1$, $K_2$, $K_3$ are candidates for category 1 ($C_1$). $K_3$, as can be seen, is closer in terms of Euclidean distance, to category 2 ($C_2$) than to $C_1$, and hence fails the Validity Filter (5). For $\{C_1, C_2\}$ as {*Baby, Clothing*}, example keywords (from actual data) for $K_1$, $K_2$, $K_3$ are '$newborn$', '$bathing$', '$shawl$' respectively. Querying '$shawl$' ($K_3$) in a video search engine is very less likely to retrieve $Baby$ related videos, than it is to retrieve $Clothing$ related videos. Thus including RV('$shawl$') in training data of category $i$ would add more mislabeled videos in T($i$) than videos having true label of category $i$. Validity Filter (5) ensures that keywords such as '$shawl$' are not valid keywords, and hence not considered to be selected as SRKs.

Let $N_{Valid,i}$ be the number of valid keywords for category $i$. We define **Suitability for Retrieving Training video** ($SRT$) score for a valid keyword $K$ as a score that indicates how suitable the valid keyword is to retrieve videos for category $i$ such that the resulting training data has the desired properties discussed in Section II. We claim that the components that lead to a high $SRT$ score for a valid keyword of category $i$, are: 1) High Proximity, 2) High Diversity. We discuss below how the above components lead to training videos having the desired properties.

***High Proximity -*** Under assumptions of whitened data and multivariate normal distributions as the categories, the likelihood of a video $v$ belonging to a category $i$ is proportional to $exp(-\frac{1}{2} \mid v - \mu_i \mid^2)$. Assuming equal prior probabilities

TABLE I: Proximity and Diversity Scores for few Valid Candidate Keywords of category Baby

| Valid Candidate Keyword | Proximity Score | Diversity score |
|---|---|---|
| alert | 0.607 | 0.3737 |
| baby carriage | 3.346 | 0.2525 |
| baby sit | 2.815 | 0.2559 |
| bassinet | 3.148 | 0.3127 |
| newborn | 1.794 | 0.3495 |
| swaddle | 3.287 | 0.1899 |
| tootsy | 0.989 | 0.4769 |
| bathing | 0.756 | 0.2036 |

for all categories, the probability that $v$ has its true label as category $i$ is higher when $\mid v - \mu_i \mid$ is lower, where $\mu_i$ is the true mean of category $i$. Thus, if $v$ is a video in training data of category $i$, then P(*v is a mislabeled video*) is lower when $\mid v - \mu_i \mid$ is lower. Consider a valid keyword $K$ for category $i$. Satisfying (5) merely implies that RV($K$) contains more videos of true label category $i$ than videos that are mislabeled as category $i$. Preference should be given to valid keywords that lead to lesser mislabeled videos in the resulting set of training videos of category $i$. In order to do so, a valid keyword $K$ should be preferred if the videos in RV($K$) are closer to $\mu_i$. We thus calculate **Proximity score** for each valid keyword $K$ as $\{1 \backslash \mid \mu_K - \mu_i \mid\}$ where $\mu_K$ is the centroid of RV($K$). A keyword $K$ should be preferred to be selected as a Selected Retriever Keyword (SRK) if the Proximity Score of $K$ is high.

In Fig. 2, '$newborn$' ($K_1$) and '$bathing$' ($K_2$) are both Valid Candidate Keywords for category $Baby$ ($C_1$). Since the videos in RV('$newborn$') are in general closer to the mean of $C_1$ than the videos in RV('$bathing$') are, the proximity score of '$newborn$' is more than that of '$bathing$'. The valid keyword '$newborn$' will thus be preferred over '$bathing$' in the proposed approach. Since *P(true label of* $v = C_1 \mid v \in RV(\text{`}newborn\text{'}))$ is more than *P(true label of* $v = C_1 \mid v \in RV(\text{`}bathing\text{'}))$, preferring '$newborn$' over '$bathing$' as SRK reduces mislabeled videos in resulting set of training videos.

***High Diversity -*** Assume that T'($i$) is the training data of category $i$. From the discussion in Section II, a valid keyword $K$ should be preferred if it leads to higher Intra-Category Diversity of $\{T'(i) \cup RV(K)\}$. We hence define **Diversity score** of a valid keyword $K$ for category $i$, given existing training data T'($i$) as $div(T'(i) \cup RV(K))$ , which can be calculated using (1).

For the set of categories {*Baby, Clothing, Fitness, Food*}, Table I shows the Proximity and Diversity scores for certain valid keywords of category $Baby$. For the purpose of calculation of these scores, the existing training data T'($i$) for category $i$ is taken to be RV($C_i$). The valid keyword '*baby carriage*' has a high Proximity score, indicating it retrieves videos that are very close (in terms of Euclidean distance) to RV('$Baby$'), but has a low Diversity score, indicating low Intra-Category Diversity of the resulting training data $\{RV(\text{`}Baby\text{'}) \cup RV(\text{`}$*Baby Carriage*$\text{'})\}$. Compared to this, '$alert$' is a Valid Candidate Keyword for $Baby$ which has a high Diversity score, but has a very low Proximity score and hence very unlikely to retrieve Baby-related training videos. Using either Proximity score or Diversity score alone leads to selection of keywords that lead to low resulting Intra-Category Diversity or high mislabeled instances in the training videos respectively. It is hence necessary to select SRKs based on both scores. Note that Table I only shows valid keywords so candidate keywords such as '$Shawl$' that fail the Validity Filter are not present.

In order to combine the Proximity and Diversity scores to obtain $SRT$ score of a valid keyword $K$, we assume $SRT$ to be a simplistic linear combination of the two scores. $SRT(K, T'(i))$ denotes the Suitability for Retrieving Training video score of a valid keyword $K$ for category $i$, given that existing training data for category $i$ is T'($i$).

$$SRT(K, T'(i)) = \alpha * \left\{ \frac{N_1}{\mid \mu_K - \mu_i \mid} \right\} + (1 - \alpha) * \left\{ N_2 . div(T(i) \bigcup RV(K)) \right\}. \quad (6)$$

Here, $N_1$ and $N_2$ are normalization factors used to ensure that Proximity and Diversity scores have the same order of magnitude in (6). $\alpha \in [0, 1]$ is the **Moderation factor**, which decides the weight given to the Proximity score relative to the Diversity score. We next discuss an iterative algorithm to obtain (a maximum of) $L$ keywords as SRK from a given set of candidates for each category, using $SRT$ as calculated in (6).

### C. SRK Selection Algorithm

The category names $C_i$, and number of SRKs to be selected ($L$) are inputs to the proposed SRK Selection Algorithm (Algorithm I) shown below. Assume that $M$ Candidate Keywords are available for each category. Let the set of Candidate Keywords for category $i$ be $K_{Candidates,i}$. For each category, a set of valid keywords $K_{Valid,i}$ is selected as a subset of $K_{Candidates,i}$ that satisfy (5). Starting with T'($i$)=RV($C_i$), $SRT(K, T'(i))$ is calculated for each valid keyword using (6), and the top keyword $K_{top}$ is selected as an SRK. T'($i$) is then updated to $\{T'(i) \cup RV(K_{top})\}$, and the process is repeated until $L$ SRKs are selected or there are no valid keywords left. The proposed algorithm selects SRKs in an iterative manner, as compared to ranking valid keywords by their $SRT$ score calculated once and selecting the top $L$ keywords. While the latter calculates $SRT$ scores independent of other SRKs selected, the proposed algorithm attempts to increase Intra-Category Diversity of the resulting training data, leading to better performance of trained classification model.

For each category, number of valid keywords may be different and hence, the number of selected SRKs by the proposed algorithm need not be similar across different categories. In order to avoid any class imbalance, we select the first $L'$ SRKs for each category to obtain training videos, where $L'$ is the minimum (across all categories) number of SRKs selected by the proposed algorithm. $L'$ may be less than $L$ since a category may have less than $L$ valid keywords. The training videos retrieved by SRKs as per (2) can be utilized to train a classification model by giving equal weight or different weights to training videos, as discussed below.

- ***Non-Weighted Instances:*** Classification model is trained by giving equal weight to all training instances (videos).
- ***Weighted Instances:*** Classification model is trained by giving a weight to a training video $v$ depending on the order in which the proposed algorithm selected the SRK corresponding to $v$. Consider a SRK $K$ that is selected for category $i$ by the proposed algorithm in the $n^{th}$ iteration. Each video $v$ in RV($K$) gets weight equal to $(1 - \frac{1}{n})$.

***Analysis of the Proposed Algorithm and Methods:*** For the two methods described above, if the candidates are distinct, selecting more SRKs is in general expected to increase the Intra-Category Diversity, thus leading to better performance. However, after certain number of SRKs are selected, it is expected that information of the new topics coming into the training data will reduce, and performance might saturate. Also, the SRKs selected in earlier iterations were determined to be more suitable for retrieving training videos than ones selected later by the proposed algorithm. The SRKs selected in last few iterations might not be very suitable to retrieve training videos of the respective category although they cleared the Validity Filter. Such SRKs might add videos of topics beyond the realm of categories of interest, and make the training data too general and less discriminative. Hence, the performance of Non-Weighted Instances method might peak at a certain $L$, and then degrade since it gives equal weight to training videos retrieved by all SRKs. In Weighted Instances method, however, the weight given to videos retrieved by SRKs accepted later is lesser. This makes the trained classifier less influenced by videos retrieved by SRKs that are selected in the last few iterations by the proposed algorithm. Thus, unlike in the case of the Non-Weighted Instances method, the performance of Weighted Instances method might just saturate with increasing $L$. The number of SRKs that lead to best classification performance for Non-Weighted Instances method may vary with the set of categories, the number and source of candidate keywords, etc. It is hence a better approach to choose as large $L$ as permitted by computational resources, and utilize the training videos for training of classification model, using Weighted Instances method.

***Complexity Analysis:*** The time-complexity of the proposed algorithm is $O(M.N_{Cat}.L)$, where $M$ is the number of candidates for each category, $N_{Cat}$ is number of categories, and $L$ is the number of SRKs selected. This is because in every iteration, the valid keywords for each category are given an $SRT$ score, and the maximum number of iterations that the proposed algorithm can run for is $L$. For fixed $M$ and $N_{Cat}$, the time-complexity of proposed algorithm varies as $O(L)$. The space-complexity of the proposed algorithm is $O(L)$ when the number of videos retrieved per SRK is kept constant. For learning tasks, when a very large sized training data is available, the size of training data used for training a model is generally constrained based on space and time-complexity of the employed learning algorithms, and available resources. Such constraints can dictate the total number of SRKs, i.e., $L$, utilized to retrieve training videos. In the following section, we provide observed space requirement, and time taken by the proposed algorithm based on our implementation, as well as its performance.

**Algorithm 1** Proposed SRK Selection Algorithm

**Inputs:** Names of categories: $C_i$,
$L$,
Candidate keywords per category: $K_{Candidates,i}$
**Initialization:** $K_{SRK,i} = [\ ](emptyset)$;
$T'(i) = RV(C_i)$
**Applying Validity Filter**
$K_{Valid,i} := K \in K_{Candidates,i} : K$ satisfies (5)
**Iterative Algorithm:**
For $n$=1 to $L$ (each iteration)
  For $i$=1 to $N_{Cat}$ (each category)
    If $K_{Valid,i}$ is empty, stop
    Calculate $SRT(K, T'(i))\forall K \in K_{Valid,i}$ as per (6)
    Select valid keyword $K_{top}$ with highest $SRT$ score
    Add $K_{top}$ to $K_{SRK,i}$; T'($i$) = T'($i$) $\cup$ RV($K_{top}$)
    Remove $K_{top}$ from $K_{Valid,i}$
  EndFor
EndFor

## IV. Experimental Results

This section discusses our experimental setup and provides performance evaluation of the proposed framework. We conduct our experiments on YouTube videos using the YouTube API. We have used Wikipedia Thesaurus API [15] and Reverse Dictionary [16] as the sources of candidate keywords given the name of a category. The candidates for a category are made distinct by removing any repetitions. The classifier used is a linear Support Vector Machine (SVM). Performance is compared against baseline classifier, which is a linear SVM trained over videos retrieved by category names. Classification accuracy is taken to be the performance measure. Textual data for a video is obtained from the Title, Keywords, and Description in the corresponding webpage. Each video webpage is represented as a bag of word vector of normalized word counts. Textual vocabulary is created based on collecting unigrams that occur in more than 0.5% of total video webpages in training data, and by removing stop words (such as *it, a, or, was* etc).

We conducted a user study to obtain videos viewed by a set of volunteers. More than 14000 videos viewed by 30 volunteers were collected. The testing videos for our proposed framework are obtained by manually labeling videos collected by the above user study. Testing videos for a category are also supplemented by videos from publicly available lists of popular (or useful or best) videos of the category.

In the next three sub-sections, we summarize results of applying our proposed approach on three different sets of categories.

### *A. Retail Product categories: Baby, Clothing, Fitness, Food*

As discussed earlier, for a retail or department store (such as Walmart or Sears), knowledge of user preferences in product categories like the above is very useful. We first discuss results

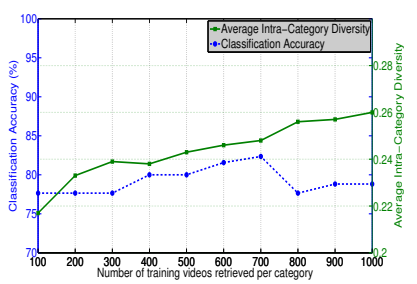


Fig. 3: Classification accuracy and Intra-Category Diversity variation with number of training videos for baseline classifier

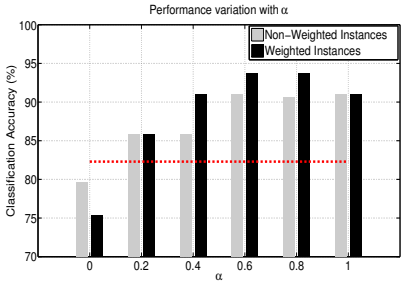


Fig. 4: Classification accuracy variation for {Baby, Clothing, Fitness, Food} with respect to $\alpha$. Dotted line shows baseline accuracy

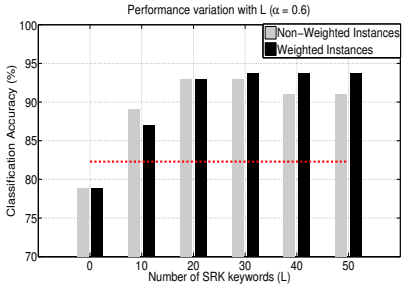


Fig. 5: Classification accuracy variation for {Baby, Clothing, Fitness, Food} with respect to L

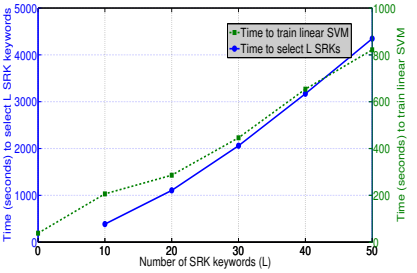


Fig. 6: Variation of time taken to select $L$ SRKs, and time taken to train linear SVM on obtained training data, with respect to $L$

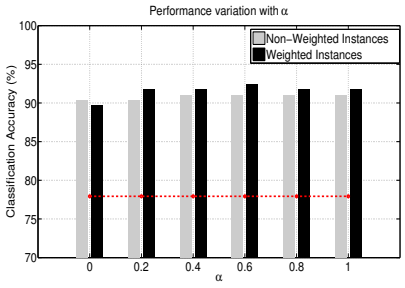


Fig. 7: Classification accuracy variation for {Classical music, Electronic Music, Jazz music, Rock music} with respect to $\alpha$

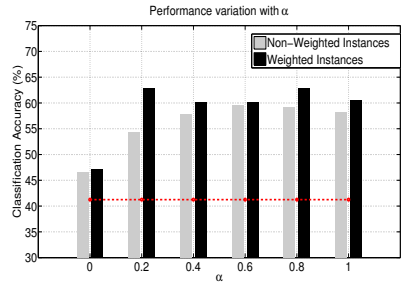


Fig. 8: Classification accuracy for {Action movies, Comedy movies, Horror movies, Romantic movies} with respect to $\alpha$

obtained using the baseline classifier followed by our proposed approach.

Fig. 3 shows the performance of the baseline classifier on classifying 255 test videos. It is seen that the performance of the baseline classifier improves initially as more videos are retrieved by category name and used for training. As shown in Fig. 3, the average (across all categories) Intra-Category Diversity of the obtained training data increases with the number of retrieved videos, initially leading to performance improvement. However, as more videos are retrieved using category name, the quality of retrieved videos by the video search engine begins degrading, and more training videos unrelated to the respective categories are retrieved and selected in their training data. This is reflected in loss in classification performance (around 700-800 videos per category) as more videos are retrieved using category name. In our experiments, 1000 videos are retrieved per SRK. In order to provide a fair comparison of our approach with the baseline, the number of training videos in both cases should be equal. However, from the trend in Fig. 3 it can be seen that the best baseline performance is around 82.3%. Since the YouTube API limits number of retrieved videos per keyword to 1000, we utilize the best classification performance (82.3%) of baseline to compare with our techniques.

In Fig. 4, we present performance of the proposed approach with varying $\alpha$. Keywords from [15] and top 200 keywords from [16] are used as candidates, giving a total of 230 candidates per category. The number of valid keywords found per category are Baby: 81, Clothing: 63, Fitness: 78, Food: 52. The coefficients $N_1$ and $N_2$ in (6) are chosen such that $N_1$=$\frac{1}{N_2}$=$div(RV(C_i))$. Fig. 4 shows the performance when $L'$ number of SRKs are selected per category, where $L'$ is the minimum number of valid keywords across all categories (which is 52 in this case). We show the performance using both Weighted Instances and Non-weighted Instances methods. Weighted support vector machine is used to give varying weights to the training videos as per the Weighted Instances method. As we can see, $\alpha$ = 0.6 to 1 performs best for Non-weighted Instances method, and $\alpha$ = 0.6 to 0.8 performs best for Weighted Instances method. We observe that the Weighted Instances method in general performs better than the Non-weighted Instances method, as we had expected in Section III. Moreover, both the methods have significantly better accuracy than the baseline classifier accuracy of 82.3% (shown by the dotted line in Fig. 4). For the baseline case, the Intra-Category Diversity values of training data are {*0.259, 0.247, 0.244, 0.243*} corresponding to {*Baby, Clothing, Fitness, Food*}. Compared to this, the Intra-Category Diversity after all 52 SRKs (for $\alpha$ = 0.6) are used to retrieve training videos for above categories are {*0.439, 0.418, 0.395, 0.359*}. The average (taken across all categories) Intra-Category Diversity has increased from **0.248** for baseline to **0.403** with our approach (for 52 SRKs per category, and $\alpha$ = 0.6). Consequently, the classifier performance has also increased from **82.3%** (baseline performance) to **91%** (Non-Weighted Instances) and **93.7%** (Weighted Instances), thus verifying that higher Intra-Category Diversity in training videos results in better performance of the trained classification model.

Fig. 5 shows how the performance of proposed framework varies with respect to $L$, i.e., the number of SRKs. $\alpha$ is kept

constant at 0.6 for the purpose of this experiment. The number of videos retrieved per keyword is 1000. As can be seen, while the performance of the Non-Weighted Instances method starts decreasing after initially increasing with increasing $L$, the Weighted Instances method performs better, and continues its improving performance with increasing $L$.

Fig. 6 shows the time taken by the proposed algorithm to select SRKs with respect to the number of SRKs selected (i.e., $L$). Conforming to the complexity discussion in Section III, the time taken to select $L$ SRKs varies as $O(L)$ when the number of candidates and categories are fixed. Fig. 6 also shows the time taken by a linear SVM (LibLinear implementation of SVM) to train over the collected set of training videos. The time-taken to learn the classifier varies approximately as $O(L)$.

For our MATLAB-based implementation, system memory usage was 4.46GBs for selecting 52 SRKs from 230 candidate keywords for category $Baby$. Training of SVM using 1000 videos for each of 52 SRKs required 1.4GBs. The empirical results presented show the feasibility of our proposed approach in terms of space and time complexities.

Next, we present experimental results for two more sets of categories. We focus on classifier performance only owing to space constraints.

### B. Genres of Music: Classical, Electronic, Jazz, Rock

We have chosen these categories keeping the requirements of a music recommendation system in mind. While there are several, and often subjective, categorizations possible within Music, we have chosen the above four categories since these cover most other categories, and are broad in the sense of ease of labeling by a human expert. Top 100 keywords from [16] are used to obtain candidates for each category. 26 SRKs are selected per category. 290 test videos are used to test the performance of both the baseline classifier and the proposed approach. In Fig. 7, we present performance with varying $\alpha$. The performance of classifier for $\alpha \geq 0.2$ is almost the same. The Weighted Instances method is again seen to outperform the Non-Weighted Instances method for higher $\alpha$ values ($\alpha \geq$0.2), and both show significantly better accuracy (92.4% and 91% respectively) than the baseline classifier (77.9%).

### C. Genres of Movies: Action, Comedy, Horror, Romantic

We here provide results for a set of categories that might be of interest for a movie recommendation system. [16] is used to obtain Candidate Keyword for each category. A total of 223 test videos are used to assess performance. 18 SRKs are selected per category. Fig. 8 shows the variation of performance of the classifier with $\alpha$. Weighted Instances is seen to be better performing than Non-Weighted Instances. Both methods show significantly improved performance (62.8% and 59.6% respectively) compared to performance of baseline classifier (41.2%).

Based on the above experiments, it can be seen that both Non-Weighted Instances and Weighted Instances methods lead to significant improvement in classification performance compared to the baseline classifier. Also, the classification performance is not very sensitive to $\alpha$ when $\alpha$ is in the range $[0.4, 1]$.

## V. Conclusions

We have proposed a fully-automated approach to obtain high quality training videos for any arbitrary set of categories, without the need for any manual labeling that is needed by most related approaches. We analyze properties of training data that lead to high performance of the trained classifier. Based on the above properties, we propose an approach for selecting keywords to retrieve training videos, on the basis of their proximity to the categories of interest, and the diversity they bring to the training data. Experimental results on several sets of categories show the effectiveness of the training videos obtained by the proposed approach, hence making classification of videos watched by users to arbitrary set of categories feasible. Consequently, this work may enable new personalization applications by enabling identification of user preferences in a set of categories relevant to the application.

*Acknowledgements*: This research was supported by the UCSD Center for Wireless Communication and the UC Discovery Grant program.

## References

[1] Nielsenwire Report, January 2010. [Online]. Available: http://blog.nielsen.com/nielsenwire/online_mobile/time-spent-viewing-online-video-up-13-in-december

[2] Y. Song, M. Zhao, J. Yagnik, and X. Wu, "Taxonomic Classification for Web-based Videos," in *IEEE Conference on Computer Vision and Pattern Recognition, 2010*.

[3] Z. Wang, M. Zhao, Y. Song, S. Kumar, and B. Li, "YouTubeCat: Learning to Categorize Wild Web Videos," in *IEEE Conference on Computer Vision and Pattern Recognition, 2010*.

[4] J. R. Zhang, Y. Song, and T. Leung, "Improving Video Classification via Youtube Video co-watch Data," in *ACM workshop on Social and Behavioural Networked Media Access, 2011*.

[5] C. Ramachandran, R. Malik, X. Jin, J. Gao, K. Nahrstedt, and J. Han, "Videomule: a Consensus Learning Approach to Multi-Label Classification from Noisy User-Generated Videos," in *ACM International Conference on Multimedia, 2009*.

[6] L. Yang, J. Liu, X. Yang, and X.-S. Hua, "Multi-Modality Web Video Categorization," in *ACM International Workshop on Multimedia Information Retrieval, 2007*.

[7] G. Schindler, L. Zitnick, and M. Brown, "Internet Video Category Recognition," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2008.*

[8] J.-F. Chen, H. Lu, R. Wei, C. Jin, and X. Xue, "An Effective Method for Video Genre Classification," in *ACM International Conference on Image and Video Retrieval, 2010*.

[9] Z. Chen, J. Cao, Y. Song, Y. Zhang, and J. Li, "Web Video Categorization based on Wikipedia Categories and Content-duplicated Open Resources," in *ACM International Conference on Multimedia, 2010*.

[10] X. Wu, C.-W. Ngo, and W.-L. Zhao, "Data-Driven Approaches to Community-Contributed Video Applications," *IEEE Multimedia, 2010*.

[11] D. Brezeale and D. J. Cook, "Automatic Video Classification: A Survey of the Literature," *IEEE Transactions on Systems, Man, and Cybernetics, 2008*.

[12] C. E. Brodley and M. A. Friedl, "Identifying and Eliminating Mislabeled Training Instances," in *Proceedings of the National Conference on Artificial Intelligence*. Citeseer, 1996.

[13] X. Zhu, X. Wu, and Q. Chen, "Eliminating Class Noise in Large Datasets," in *Machine Learning International Workshop, 2003*.

[14] K. Fukunaga and D. R. Olsen, "An Algorithm for Finding Intrinsic Dimensionality of Data," *IEEE Transactions on Computers, 1971*.

[15] K. Nakayama, T. Hara, and S. Nishio, "Wikipedia Mining for an Association Web Thesaurus Construction," in *Web Information Systems Engineering, 2007*. Springer.

[16] Reverse Dictionary. [Online]. Available: http://www.onelook.com/reverse-dictionary.shtml