Learning to Rank Videos Personally Using Multiple Clues

Songhua Xu[♮,♯,‡*]

♮: College of Computer Science and Technology, Zhejiang University, Hangzhou, Zhejiang, 310027, P.R. China

Hao Jiang[‡]

♯: Department of Computer Science, Yale University, New Haven, Connecticut, 06520-8285, USA

Francis C.M. Lau[‡]

‡: Department of Computer Science, The University of Hong Kong, Pokfulam Road, Hong Kong, P.R. China

ABSTRACT

In this paper, we introduce a new learning based video content similarity model. The model leverages on multiple clues on the contents of a video and can be used to rank videos in a personalized way. The key to produce a personalized video ranking is to have a good estimate of pairwise video content similarity, which is realized through meta-learning using a radial-basis function network. Four aspects of a video are considered in deriving the video content similarity in our method. The training data to our model are acquired in the form of user judged preference relationships regarding video content similarities. With the optimized video content similarity estimation obtained by our algorithm, we can produce a personalized video ranking that matches more closely an individual user's watching interest over a collection of videos. The video ranking results generated by our prototype system are compared with the groundtruth rankings supplied by the individual users as well as rankings by the commercial video website YouTube. The results confirm the advantages of our method in generating personalized video rankings.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Relevance feedback; H.3.7 [**Digital Libraries**]: User issues; H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Algorithms, Design, Experimentation, Human Factors, Measurement, Performance

Keywords

Personalized video ranking, learning to rank videos, video similarity estimation, multi-modality video similarity fusion, human factors in information retrieval, user feedback

*Contact him at A DOT B AT C DOT com in which A = "songhua", B = "xu", and C = "gmail".

CIVR '09, July 8-10, 2009 Santorini, GR

1. INTRODUCTION

Online videos are in great abundance on the Internet. At present, users find videos from the Internet mostly through keyword based search which returns a uniform set of search results for all the users. This is not most desirable because video watching interests vary from person to person and ideally such search results should be ranked according to the user's interests. In our prior study [28], we have proposed an example learning based algorithm to address such a need of personalized video recommendation. One key step involved in that algorithm is to perform estimation of pairwise video content similarities. In this paper, we propose a new video similarity model which can estimate more reliably pairwise video content similarity. With this refined video similarity estimation, we can produce a more accurate personalized video ranking to approximate individual user's video watching interests. Since a good video content similarity estimation plays a critical role in content based video retrieval, our work here provides a useful reference for related efforts in system design for content-based video retrieval.

Our new video similarity model makes use of meta-learning of video similarities in four aspects of a video: video visual similarity, description text similarity, audio similarity and video text similarity. These four kinds of video similarities can be derived using existing algorithms. Given these base video content similarities, we introduce a meta-learning based process to synthesize an overall similarity measure which fuses together these partial video similarity estimates. To make the training data labeling process more user-friendly, affordable and reliable, we adopt preference relationships to represent relative video similarities between pairs of training videos.

The structure of this paper is as follows. Section 2 discusses some related work. Section 3 explains our novel content-based video similarity estimation based on meta-learning. Section 4 presents our main algorithm for personalized video ranking through predicting the user's individual video watching interest based on the optimized video content similarity estimation. Section 5 reports some selected experiment results and Section 6 concludes the paper.

2. RELATED WORK

As mentioned earlier, the key to personalized video ranking is to estimate video content similarity in the eyes of a certain user. These estimates can then be used to predict the user's personal video watching interest. There exists many algorithms for estimating video content similarity. In the following, we briefly look at some of the most related studies.

Tan et al. [25] proposed a dynamic programming based framework for measuring video similarity and successfully applied their method to video query by example. Their dynamic programming

framework optimally aligns individual frames between two videos being compared. Wu et al. [27] proposed a video similarity model which can make adjustments based on the opinions of different users via a relevance feedback mechanism. In their method, candidate videos that achieve the highest content similarity scores are singled out and the users are asked to manually assign scores to a subset of these videos. These scores are then used to adaptively optimize the internal parameters in the video similarity estimation model. Cheung and Zakhor [5] proposed a video signature concept for measuring video content similarity where they define the distance between a pair of videos as the average distance between the best matched pairs of individual frames in the two videos. Using the video signature concept, they successfully detected multiplicity of videos on the Internet [6]. In a follow-up work, they tried retrieving videos through clustering the video signatures [7], which resulted in improved retrieval precision and accuracy. The clustering procedure is applied to the lengths of edges in a minimum spanning tree which is derived from a graph constructed based on video signatures. Lin et al. [17] integrated color and spatial features to estimate video similarity and proposed the concepts of dominant color histogram and spatial structure histogram for representing visual content variations in a video. For better accuracy, they divide a video into multiple subshots through analyzing visual content coherence between adjacent video frames. The overall video similarity is then estimated according to the best corresponding subshots. Experiment results show that their method can improve average recall in video retrieval. Manjunath et al. [19] studied the low-level color and texture descriptors useful for video similarity estimation and content based video retrieval. Kim and Park [12] introduced the modified Hausdorff distance and used the directed divergence method to efficiently match video sequences for more accurate estimation of video similarity. Hoi et al. [10] proposed a hybrid scheme for efficiently evaluating video similarity by utilizing video signatures on multiple granularities. The coarse signature is derived according to pyramid density histograms, which is used to first filter out most of the videos with poor similarities; and then a fine signature is estimated via nearest feature trajectory for more refined video content similarity comparison. Their algorithm is particularly suited for applications that retrieve duplicate video copies, e.g., to detect videos on the Internet without proper copyright permission. Liu et al. [18] proposed a video content signature based on video histogram analysis. The video signature they derive belongs in a low dimensional space, which facilitates efficient detection of duplicated copies of online videos.

All the above video content similarity estimation algorithms are based on visual similarities. Cheung and Zakhor [5] introduced a meta-data based method to detect similar copies of online videos according to a video's associated hyperlink information and its descriptive texts, e.g., the authors, copyright and title information. Senechal et al. [23] proposed a hybrid audio-visual signature including both an audio signature and a spatio-temporal video signature. They demonstrated that their hybrid method works more robustly than audio-only and video-only signatures. Ahmad et al. [1] explored the possibility of using audio-based queries for retrieving audio-visual videos by solely looking at the audio contents of a video. Their method is particularly suited for retrieving music and speech videos. In general, video content similarity estimation utilizing audio data is closely related to audio similarity estimation for audio retrieval applications, for which a fair amount of work exists. On that area, audio fingerprinting based audio retrieval has recently become a hot research issue [4, 20, 14].

3. VIDEO SIMILARITY ESTIMATION VIA META-LEARNING

There exist many algorithms for estimating video similarity, as have been surveyed in Section 2. However, none of these methods appear to be suitable for reliably estimating video similarity under a generic setting for all possible circumstances. Here we propose a novel meta-learning based video similarity algorithm. In constructing our meta-learning algorithm, we utilize existing content similarity algorithms to measure the following four kinds of similarities of a video: 1) visual content similarity, 2) description text similarity, 3) audio similarity, and 4) video text similarity. By fusing together these different video similarities through meta-learning, we obtain a video similarity estimation which can more reliably and comprehensively indicate the content similarity between a pair of videos. In the following, we look at the algorithms for estimating these different kinds of video content similarity, in Section 3.1–Section 3.4. After that, in Section 3.5, we introduce our new video similarity estimation algorithm which is based on meta-learning of the partial video similarity estimations in the above four aspects of a video through a radial-basis function network.

3.1 Estimating Visual Content Similarity

To estimate the visual content similarity between a pair of videos, we employ two existing algorithms to compute the similarity between the corresponding frames of two videos—the one in [8] measures the similarity using the video signature concept, and the other is the content-based video similarity model proposed in [27]. We refer to these two algorithms as VC_1, VC_2 respectively.

3.2 Estimating Description Text Similarity

Many online videos include some description texts which summarize or highlight the main contents of a video. These description texts serve as a major clue for users to decide whether to watch a clip of online video or not, and is a key feature used in current text-based video search engines.

There exist many algorithms for estimating pairwise text similarity. A number of similarity metrics are listed in *http://www.dcs.shef.ac.uk/%7Esam/stringmetrics.html*, each of which has its own merits and strengths for handling a certain type of texts. In our current method design, we have chosen the following five text similarity methods: the cosine based text similarity estimation, the Jaccard method for text similarity estimation, the extended Jaccard method (Tanimoto) for text similarity estimation, the Euclidean distance based text similarity estimation, and the Dice's coefficients based text similarity estimation. The implementations of these algorithms are provided by the "Simpack" open source package [2, 29] (can be freely downloaded from *http://www.ifi.unizh.ch/ddis/simpack.html*). In the later part of this paper, we refer to these algorithms as DT_1, $\cdots$, DT_5 respectively.

3.3 Estimating Audio Similarity

Prior research in the field of audio analysis, retrieval and classification has resulted in many algorithms for measuring content similarity between a pair of audio clips. Some of them are generically applicable to all types of audio data while the others are for audio in specific domains and/or for specific applications such as music genre classification and voice recognition. In our current work, we have selected three kinds of audio similarity measures as suggested in [26], [21] and [13] respectively. The paper [26] presents an audio classification algorithm based on audio content similarity, whose implementation has been made available by the authors. The audio content similarity model proposed in [21] measures content distances between a pair of audio clips. Their model is proposed

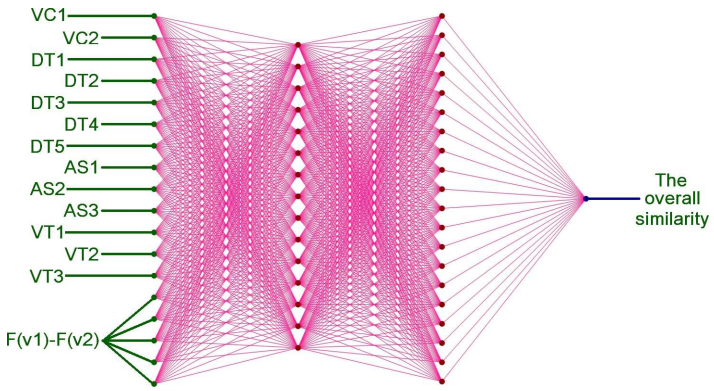

Figure 1: Our radial-basis function network for estimating pairwise video similarity.

for processing audio with generic contents, which is the main reason for its inclusion in our system. The third pairwise audio similarity algorithm [13] was proposed for a music search engine. We include this similarity model to make our overall method also capable of handling music audio data. In the later part of this paper, we refer to these three audio similarity algorithms as AS_1, AS_2, AS_3 respectively.

3.4 Estimating Video Text Similarity

Many videos have some texts in their visual contents, which we call video texts. The prior study conducted by Lienhart and Effelsberg [16] has shown the advantages of using video texts for video indexing. In this paper, we also utilize video texts for pairwise video similarity estimation. We use three existing algorithms to extract texts from a video: [24, 15, 16]. We implemented the first two algorithms ourselves while the implementation of the last algorithm was provided by the authors. Once video texts have been extracted from the two input videos using one of the above algorithms, we can estimate their similarity as the percentage of overlapping video texts in the two videos. We denote the above three video similarity estimation algorithms based on video text as VT_1, VT_2, VT_3 respectively.

3.5 Radial-Basis Function Network For Estimating Pairwise Video Similarity

Given a pair of videos v_i and v_j, using the algorithms discussed above (Section 3.1–Section 3.4), we derive a set of pairwise video content similarities. To recall, for a pair of videos we have derived their visual content similarities $VC_1(v_i, v_j)$, $VC_2(v_i, v_j)$, their description text similarities $DT_1\ (v_i, v_j)$, $\cdots$, $DT_5(v_i, v_j)$, their audio similarities $AS_1(v_i, v_j)$, $\cdots$, $AS_3(v_i, v_j)$ and their video text similarities $VT_1(v_i, v_j)$, $\cdots$, $VT_3(v_i, v_j)$ respectively in the above processing. These 13 similarity estimates are treated as the base video similarity estimates. In our new method for estimating pairwise video similarity, we introduce a radial-basis function network to derive an overall pairwise video similarity estimate using all these base similarity estimates. The scheme of the radial-basis function network model is illustrated in Figure 1, which is a classical multi-layer perceptron with two hidden layers. In the following, we first look at how to derive the input to the radial-basis function network, and then examine how to train the network.

3.5.1 Preparing the input for the network

The input to our radial-basis function network is in a 18 dimensional space, which includes all the 13 aforementioned similarity estimates as well as a 5 dimensional vector indicating the difference between the signatures of the two input videos.

We first look at how to derive a signature for an input video. For all the video clips in our local video repository, we first derive a time averaged image for each video. That is, each pixel of the resultant image is the weighted average of all the pixel values at the image location across all the frames in the video. For simplicity, we use the RGB space in this average image derivation process even though more sophisticated color spaces can also be employed. With all the average images derived, one for each video, we adopt the image clustering algorithm based on non-negative matrix factorization [15] to cluster all these average video images into five clusters. Here we define the distance between two images as the sum of the differences between the RGB values of all the corresponding pixels in these two images. For each resultant image cluster, we identify the cluster's central image I_{center} as the image which minimizes the pairwise image distance with all the other images in the cluster, i.e.:

$$I_{\text{center}} = arg\min_{I_z} \sum_k \Big(\sum_{i,j} |I_k(i,j).r - I_z(i,j).r| + \sum_{i,j} |I_k(i,j).g - I_z(i,j).g| + \sum_{i,j} |I_k(i,j).b - I_z(i,j).b| \Big), \quad (1)$$

where $I_k(i,j).r, I_k(i,j).g, I_k(i,j).b$ are the R, G, B channels of the pixel in the k-th image in the cluster with the coordinate (i,j). In this way we derive five center images, one for each resultant image cluster computed above. Inspired by [5, 8], we define the video signature as a five dimensional vector. To derive the signature, for each frame in the video, we find its closest image among the five center images. Here we also use the above sum-of-pixel-difference as the image distance measurement. Assume there are a total of n frames in a video v, among which there are n_l frames appearing closest to the l-th center image where $l = 1, 2, \cdots, 5$. Then v's video signature $\mathbf{F}(v)$ is defined as:

$$\mathbf{F}(v) \triangleq (\frac{n_1}{n}, \frac{n_2}{n}, \cdots, \frac{n_5}{n}). \quad (2)$$

Once the two video signatures are derived, one for v_i and the other for v_j, their difference vector $\mathbf{F}(v_i) - \mathbf{F}(v_j)$ can be computed, which will be used as the last 5 dimensions of the 18 dimensional input for our radial-basis function network. The output of our network is designed to be the overall video content similarity between v_i and v_j.

3.5.2 Training the network

The output of our radial-basis function network is the overall similarity between a pair of videos, which is a single number in the range of $[0, 1]$. However, obtaining reliable human labeling over a

number of pairwise video similarities is non-trivial. The subjects participating in our experiments generally felt it is more intuitive to specify the relative similarities between pairs of videos rather than giving numerical scores on the similarities of pairs of videos. This shows that to acquire an ordered list or partially ordered list of videos in terms of their content similarities is more feasible and reliable than asking participants to provide numeric values for pairwise video similarities in the range of $[0, 1]$. We notice even if we only ask our subjects to assign discrete levels on pairwise video similarities, most users still feel it is too demanding to give a consistent rating even over a small collection of videos. Inspired by the "social choice function" [22] which uses preferential relationship to represent a social choice, we use preferential relationship for representing relative video content similarities.

Formally, each record in our training data collection is concerned with three videos v_1, v_2 and v_3. Among them, if a user feels v_2 is more relevant to v_1 than v_3, we represent this user supplied groundtruth information on relative video content similarity as $\vartheta(v_1, v_2) \succ \vartheta(v_1, v_3)$, or $\vartheta(v_1, v_2) \prec \vartheta(v_1, v_3)$ if the user indicates the opposite. In case he feels v_2 is equally relevant (or irrelevant) to v_1 than v_3, we represent the information as $\vartheta(v_1, v_2) \approx \vartheta(v_1, v_3)$. In the following, we first introduce a graph theory based method to determine the optimal set of video pairs to present to a user for acquiring the groundtruth video similarity labeling from the user. We will then explain how to derive the error of our radial-basis function network given the user supplied groundtruth data.

A. Optimally determining training video pairs

We introduce a graph theory based approach to optimally determine which video pairs to ask users to label when collecting our training data. This is because exhaustively asking the users to label all the possible video pairs is impractical even for a training video set with modest size. Assuming we have n videos in our training set, denoted as $v_1, \cdots, v_n$ respectively, there are $m = \frac{n(n-1)}{2}$ enumerations of all the possible video pairs, denoted as $P(v_{i_1}, v_{j_1})$, $\cdots$, $P(v_{i_m}, v_{j_m})$ respectively. For each such video pair, we introduce a vertex in our graph which initially contains no vertices or edges. At the beginning, we randomly pick three videos and ask the user to label which two videos are most similar. Without loss of generality, we assume these three videos are v_1, v_2 and v_3 and the user labeling result is either $\vartheta(v_1, v_2) \succ \vartheta(v_2, v_3)$ or $\vartheta(v_1, v_2) \prec \vartheta(v_2, v_3)$ or $\vartheta(v_1, v_2) \approx \vartheta(v_2, v_3)$. If the user labels $\vartheta(v_1, v_2) \succ \vartheta(v_2, v_3)$, we would draw a directed edge in our graph from the vertex $P(v_1, v_2)$ to the vertex $P(v_2, v_3)$; if he labels $\vartheta(v_1, v_2) \prec \vartheta(v_2, v_3)$, we would draw a directed edge from the vertex $P(v_2, v_3)$ to the vertex $P(v_1, v_2)$; if he labels $\vartheta(v_1, v_2) \approx \vartheta(v_2, v_3)$, we would draw both edges, one from $P(v_1, v_2)$ to $P(v_2, v_3)$ and the other from $P(v_2, v_3)$ to $P(v_1, v_2)$. After that, we compute the shortest distances from each vertex P_a in the graph to every other vertex P_b in the graph, using the Floyd-Warshall algorithm [9]. We denote the distance from the vertex P_a to the vertex P_b as $Dis(P_a, P_b)$. If the resultant distance is not infinite, it means by applying transitivity and given the information the user provided so far, our system can automatically infer that the pair of videos denoted by P_a are more similar than the pair of videos denoted by P_b. The path from P_a to P_b becomes the actual inference chain. For example, if there exists a path from P_a to P_c and then from P_c to P_b, we know there exists the following preference relationships: $P_a \succ P_c$ and $P_c \succ P_b$. Thus by applying forward inference, we know $P_a \succ P_b$, which is what is actually represented by the path from P_a to P_b. If both $Dis(P_a, P_b)$ and $Dis(P_b, P_a)$ are infinite, it means neither there exists a path from P_a to P_b, nor a path in the reverse direction. In our context, this means by applying transitivity, we cannot infer any relative video content similarity relationship between the video pairs P_a and P_b. Based on the above approach, to determine the optimal set of video triplets to ask the user to label in the training example acquisition process, we find all the vertex pairs whose shortest connection path is the longest. This means either the user's labeling data so far do not allow our system to automatically infer which pair of videos are more similar in contents than the other pair, or alternatively our system can infer their similarity but through the largest number of steps. Both situations mean user labeling over the similarity between these pairs of videos is more informative and revealing than labeling the rest of the video pairs. Among these vertex pairs, each time we randomly select two pairs of videos and ask a user to label the relative similarities between the videos in these pairs, i.e., whether the videos in one pair are more similar in their contents than videos in the other pair. During the random selection process, we give priority to those video pairs which share one video in common, because this can facilitate mental video content similarity comparison during the user labeling process. With such optimal training video pair selection method, we can always get the most useful information from a given amount of user labeling effort.

B. Determining the error of the prediction network

Given a training set of relevance relationships as collected above, we can derive the error of a radial-basis function network under a certain configuration. The details are as follows. Assume the training set specifies query relevance relationships over m pairs of videos and these pairs involve n videos $\mathbf{V} = \{v_1, \cdots, v_n\}$. We denote the video pair set $\mathbf{VP} = \{\vartheta_k(v_{i_k}, v_{j_k}),\ v_{i_k},\ v_{j_k} \in \mathbf{V},\ v_{i_k} \neq v_{j_k},\ 1 \leq k \leq m\}$. For all the video pairs in $\mathbf{VP}$, we first use our radial-basis function network under its current configuration to predict pairwise video similarities. This network can be used to produce a partial order between the videos in terms of their content similarities. And then we use the above graph algorithm to find the shortest distances among these m video pairs, based on which we derive an overall error E between our radial-basis function network output and the user specified video relevance to the query. For each pair of video pairs $(\vartheta_a, \vartheta_b)$, if the shortest path from ϑ_a to ϑ_b is not infinity, we know $\vartheta_a \succ \vartheta_b$. Let the path length be $\eta(\vartheta_a, \vartheta_b)$. To derive the accumulated error term E, we use the following method: If the network falsely predicts a relationship of $\vartheta_a \succ \vartheta_b$ to be $\vartheta_a \prec \vartheta_b$ or vice versa, we would increase E by one. If the network mistakenly predicts the relationship of $\vartheta_a \approx \vartheta_b$ to be $\vartheta_a \succ \vartheta_b$ or $\vartheta_a \prec \vartheta_b$, we would increase the accumulated error E by 0.5. Because not all the similarity relationships from the user specified data have the same level of confidence, we modulate the accumulated error with the distance of the path. The rationale behind is that the longer the path is, the more steps of inference we have to go through in deducing the user specified video relevance relationship, and the less reliable the result would be. So we divide the local error by the length of the path. We denote the relationships predicted by our radial-basis function network as $\prec^n, \succ^n, \approx^n$ and the relationships deducted from the graph-based method above according to the user supplied data as $\prec^u, \succ^u, \approx^u$. Then we can mathematically state the above process as:

$$E(\vartheta_a, \vartheta_b) \triangleq \tag{3}$$

$$\begin{cases} \frac{1}{\eta(\vartheta_a, \vartheta_b)} & \text{if } (\vartheta_a \prec^n \vartheta_b \wedge \vartheta_a \succ^u \vartheta_b); \\ \frac{1}{\eta(\vartheta_a, \vartheta_b)} & \text{if } (\vartheta_a \succ^n \vartheta_b \wedge \vartheta_a \prec^u \vartheta_b); \\ \frac{0.5}{\eta(\vartheta_a, \vartheta_b)} & \text{if } ((\vartheta_a \succ^n \vartheta_b \vee \vartheta_a \prec^n \vartheta_b) \wedge \vartheta_a \approx^u \vartheta_b); \\ \frac{0.5}{\eta(\vartheta_a, \vartheta_b)} & \text{if } (\vartheta_a \approx^n \vartheta_b \wedge (\vartheta_a \succ^u \vartheta_b \vee \vartheta_a \prec^u \vartheta_b)); \\ 0 & \text{otherwise.} \end{cases}$$

And the overall error is:

$$E \triangleq \sum_{(\vartheta_a,\vartheta_b)\in \mathbf{VP}} E(\vartheta_a,\vartheta_b). \quad (4)$$

Given this error measurement, we then train our radial-basis function network using a modified version of the genetic algorithm as proposed in [3]. The only difference in our modified training process is that we derive the error of the network through the above error measurement process. To avoid over-fitting in the training process, we employ the 10-fold cross-validation technique. That is, with our training set of videos whose relative video content similarities are known, we first randomly pick 90% of the videos as the training samples and leave the remaining 10% as the testing samples, and we report the whole process for ten times. Here the training video set consists of over 3000 video clips downloaded from YouTube.

4. PERSONALIZED VIDEO RANKING BASED ON VIDEO SIMILARITY

4.1 Main Idea

Given our new pairwise video content similarity estimation model, we can now develop a personalized video ranking method based on the video content similarity estimation. The personalized video ranking is essentially produced by first predicting individual users' video watching interests over all the videos in a video query result set which is obtained by a conventional text-based video search engine. Once a user's video watching interest has been predicted, we can then rank the videos according to the user's watching interest.

Our user video watching interest prediction is based on the pairwise video content similarities. We assume if the contents of two videos are sufficiently similar, then a user shall have more or less the same amount of interest to play either of them, whichever appears to the user first. We use $Sim(v_0, v_1)$ to denote the overall similarity between video v_0 and video v_1, where $Sim(v_0, v_1) \in [0, 1]$. We argue correct estimation on $Sim(v_0, v_1)$ plays a critical role in our user interest prediction. In our experiment result section, we look at how different video similarity metrics would affect the performance of personalized video ranking.

We denote the training sample set as $\{t_{watch}(u, v_i)|i = 1, \cdots, n\}$ where n is the number of videos the user u has watched so far; these videos are denoted as v_i $(i = 1, \cdots, n)$. When a new video v_x arrives, we calculate the similarity between v_x and all the videos in the training set. We then select k videos which have the highest similarity with v_x using a video similarity metric. In our current experiment, k is set as $min(10, n)$, where n is the size of the current training sample set. Without loss of generality and for ease of notation, we assume they are the videos v_i $(i = 1, \cdots, k)$. Then we use a simple linear interpolation to predict the potential video watching time of the user over v_x:

$$t_{watch}(u, v_x) = \frac{\sum_{i=1}^{k} t_{watch}(u, v_i) Sim(v_i, v_x)}{\sum_{i=1}^{k} Sim(v_i, v_x)}. \quad (5)$$

The above personalized video ranking method based on pairwise video content similarity is suggested in [28]. Interested readers can refer to that paper.

4.2 Implementation

To conduct the experiment, we develop a prototype web search interface, which consists of a client side for acquiring the video watching time of individual users and a server side for producing the user-oriented video ranks based on the prediction of watching times of users on videos they have not yet watched.

On the client side, our customized web browser plugin for acquiring user video watching time is developed as a Firefox extension plugin. This plugin records the duration that a piece of online video is actively played to a user and periodically sends the measured watching time records to the server side.

The server side implements a search engine using Java. When the server side application receives a search query submitted by a certain user, the application will forward the query to YouTube first and fetch the first 300 records if they have not been previously downloaded locally. Then our search engine predicts the watching time of the user over each such video through (5), if the watching time of the user over the video is unknown.

5. EXPERIMENT RESULTS

To evaluate the quality of the personalized video ranking generated by the algorithm introduced in this paper using our new video content similarity estimation, we conducted the following evaluation experiments. In each session of our experiment, we use our customized web browser plugin to acquire individual users' video watching times. Only the top 20 videos are used in each of our evaluation tests since they usually represent the most relevant ones to the query and also represent an affordable amount for users to manually specify their preferred video rankings. 80 users from our university are invited to participate in these evaluation experiments. For each participant, he or she is asked to watch the first 20 videos returned from YouTube (sorted by relevance) on a given query. After that, the user is asked to provide a video ranking over these 20 videos. Each participant is asked to conduct the experiment for 5 different queries. We then use our personalized video ranking algorithm to generate the video ranking for these 20 videos using the watching time data of only the first κ videos that the user has watched for $\kappa = 1, \cdots, 10$. Our algorithm then generates personalized video ranking for all the 20 videos based on the training data from the corresponding participating user. We compare both YouTube video ranking and our algorithm's video ranking result with the groundtruth video ranking supplied by the user. During our evaluation, we measure the similarity between the two video rankings using the Kendall's Tau coefficient which was introduced in [11] for measuring the agreement of two ordered lists for information retrieval applications.

Table 1 shows the statistics of 20 personalized video ranking experiments we conducted following the above setting. Each video query experiment was repeatedly conducted by 20 users separately. In total, we have $20 \times 20 = 400$ independent video querying sessions contributed by these 80 participants. The text queries used for the 20 video queryinig experiments are "April fool's day", "Angkor wat", "Argentina", "BBC", "biohazard", "butterfly effect", "champion league", "cooking school", "earthquake", "Java", "jingle bells", "Kung Fu", "national geography", "the matrix", "rocket", "Ronaldo", "Shangrila", "sky diving", "winning eleven", and "yesterday once more" respectively. Figure 2 provides a corresponding boxplot diagram to illustrate these performance statistics. In the figure, we employ the boxplots (box-and-whisker diagrams) which can more comprehensively report the statistic distribution of user performance data. The y-axis shows the corresponding Kendall's Tau coefficients of video ranks generated by our personalized video ranking algorithm (see Section 4) when using different video similarities in the ranking process. The value range for the Tau coefficients is $[-1, 1]$. Figure 2.(a) shows the performance of the initial YouTube video ranking with respect to the user expected ideal video ranking. These performance data are used as the base line in

		(a)	(b)	(c)	(d)	(e)	(f)	(g)
(I)	Avg	-0.26	-0.26	-0.30	-0.30	-0.30	-0.35	-0.36
	Min	-0.46	-0.46	-0.40	-0.47	-0.47	-0.46	-0.45
	25%	-0.35	-0.30	-0.36	-0.38	-0.37	-0.39	-0.40
	75%	-0.18	-0.14	-0.23	-0.18	-0.17	-0.32	-0.30
	Max	-0.05	-0.01	-0.12	-0.07	-0.10	-0.23	-0.20
(II)	Avg	-0.11	-0.08	-0.23	-0.21	-0.15	-0.29	-0.29
	Min	-0.40	-0.37	-0.40	-0.40	-0.39	-0.40	-0.56
	25%	-0.19	-0.19	-0.28	-0.24	-0.25	-0.33	-0.35
	75%	-0.04	0.01	-0.12	-0.17	-0.10	-0.24	-0.22
	Max	0.05	0.30	0.03	-0.01	0.02	-0.13	-0.03
(III)	Avg	-0.07	-0.02	-0.14	-0.08	-0.04	-0.26	-0.24
	Min	-0.28	-0.23	-0.40	-0.34	-0.30	-0.40	-0.56
	25%	-0.18	-0.13	-0.20	-0.18	-0.13	-0.32	-0.31
	75%	0.06	0.08	-0.04	-0.02	0.08	-0.16	-0.15
	Max	0.15	0.35	0.21	0.19	0.19	-0.04	-0.01
(IV)	Avg	0.01	0.12	-0.06	-0.02	0.08	-0.17	-0.16
	Min	-0.29	-0.23	-0.31	-0.34	-0.29	-0.40	-0.62
	25%	-0.08	0.00	-0.18	-0.07	-0.07	-0.25	-0.24
	75%	0.10	0.20	0.09	0.09	0.12	-0.13	-0.07
	Max	0.33	0.43	0.20	0.22	0.27	0.05	0.09
(V)	Avg	0.13	0.24	0.02	0.06	0.08	-0.13	-0.16
	Min	-0.20	-0.08	-0.34	-0.34	-0.26	-0.37	-0.62
	25%	0.04	0.13	-0.07	-0.02	0.01	-0.23	-0.24
	75%	0.21	0.33	0.12	0.16	0.17	-0.09	-0.03
	Max	0.38	0.47	0.25	0.31	0.34	0.13	0.30

		(a)	(b)	(c)	(d)	(e)	(f)	(g)
(VI)	Avg	0.19	0.31	0.08	0.12	0.20	-0.06	-0.07
	Min	-0.20	-0.03	-0.31	-0.26	-0.11	-0.37	-0.67
	25%	0.11	0.17	-0.03	0.01	0.10	-0.18	-0.22
	75%	0.36	0.41	0.23	0.22	0.30	0.01	0.06
	Max	0.52	0.65	0.41	0.45	0.54	0.24	0.28
(VII)	Avg	0.33	0.44	0.19	0.24	0.27	-0.06	-0.03
	Min	-0.17	-0.06	-0.31	-0.26	-0.06	-0.37	-0.66
	25%	0.24	0.33	0.10	0.09	0.15	-0.19	-0.20
	75%	0.43	0.53	0.25	0.33	0.43	0.07	0.09
	Max	0.57	0.64	0.43	0.57	0.66	0.30	0.26
(VIII)	Avg	0.48	0.49	0.21	0.33	0.40	-0.02	0.05
	Min	0.00	0.14	-0.29	-0.20	-0.06	-0.37	-0.71
	25%	0.36	0.44	0.09	0.22	0.25	-0.16	-0.16
	75%	0.59	0.65	0.39	0.42	0.54	0.14	0.18
	Max	0.75	0.83	0.63	0.59	0.71	0.37	0.44
(IX)	Avg	0.60	0.69	0.32	0.44	0.46	0.02	0.04
	Min	-0.06	0.14	-0.26	-0.23	0.00	-0.37	-0.78
	25%	0.44	0.58	0.17	0.28	0.36	-0.13	-0.15
	75%	0.66	0.77	0.44	0.51	0.59	0.15	0.27
	Max	0.78	0.84	0.64	0.72	0.86	0.43	0.49
(X)	Avg	0.66	0.73	0.36	0.50	0.55	0.09	0.11
	Min	0.06	0.23	-0.26	-0.20	0.00	-0.37	-0.72
	25%	0.54	0.63	0.23	0.34	0.46	-0.11	-0.10
	75%	0.72	0.81	0.52	0.62	0.66	0.25	0.33
	Max	0.82	0.95	0.65	0.79	0.82	0.44	0.62

(a) Description (b) Visual+Description+Audio+Text (c) Visual (d) Viusal+Audio (e) Visual+Audio+Text (f) Text (g) Audio

Table 1: Statistics on the Kendall's Tau coefficients as measurement over the qualities of 20 personalized video ranking experiments using 20 different video search keywords. (I)–(X) respectively represent the Tau values on the qualities of the personalized video ranking generated using the training data after the users have watched the first one, two, and up to ten videos during our experiment. (a)–(g) explore the effect of using different video similarity estimation methods in our personalized video ranking method. Here "Description" stands for description text based similarities (DT), "Visual" stands for visual content based similarities (VC), "Audio" stands for audio based similarities (AS), and "Text" stands for video text based similarities (VT). A boxplot diagram illustrating these data is also available in Figure 2.

our comparison. To study the effectiveness of our new video similarity model for personalized video ranking, in each experiment we evaluate the video ranking results generated using different video similarity metrics: visual content similarity, audio content similarity, video text similarity, and description text similarity, as well as combined video similarities estimated by our meta-learning based video similarity model. For comparison purpose, we also examine two types of intermediate video similarity combination options: the option of combining visual and audio content similarities together, and the option of combining visual, audio and video text similarities together.

From all the experiment results we obtained, it can be seen clearly that using the meta-learning based video similarity estimation approach through leveraging video content similarities from multiple clues, in the majority of the cases, our personalized video ranking algorithm produces personalized video ranking in better agreement with the user desired ideal video ranking. For both the annotated (with text) and un-annotated (without text) videos, our meta-learning based video similarity model can significantly improve the user video search experience. Especially for un-annotated videos, we can achieve a ranking result that is close to those for annotated videos by leveraging all types of video similarities, which is very encouraging.

In summary, according to the results of the experiments reported above, we have verified that our new video similarity model can indeed help generate personalized video rankings that are more reflective of a user's personal video watching interest as compared with existing methods. We expect users can experience a significant saving in video browsing and searching time when finding their favored clips by using our proposed personal video ranking method.

6. CONCLUSION

In this paper, we propose a new video similarity model based on meta-learning of a number of existent video similarity estimations for personalized video ranking. Since our new video similarity model is a learning based method, we also introduce a novel way to optimally acquire user labeling data as well as an error measurement method for our radial-basis function network during its training stage. The reported statistics clearly show that our video similarity algorithm can satisfactorily produce a new personalized video ranking in better agreement with the user's expectation, watching interest and preference, as verified by a comparison against the benchmark algorithm by YouTube. These results also show the potential of a new type of personalized web search service based on the algorithm suggested in this paper.

In the future, we intend to improve the accuracy of the video similarity metrics by incorporating user feedbacks on the fly through

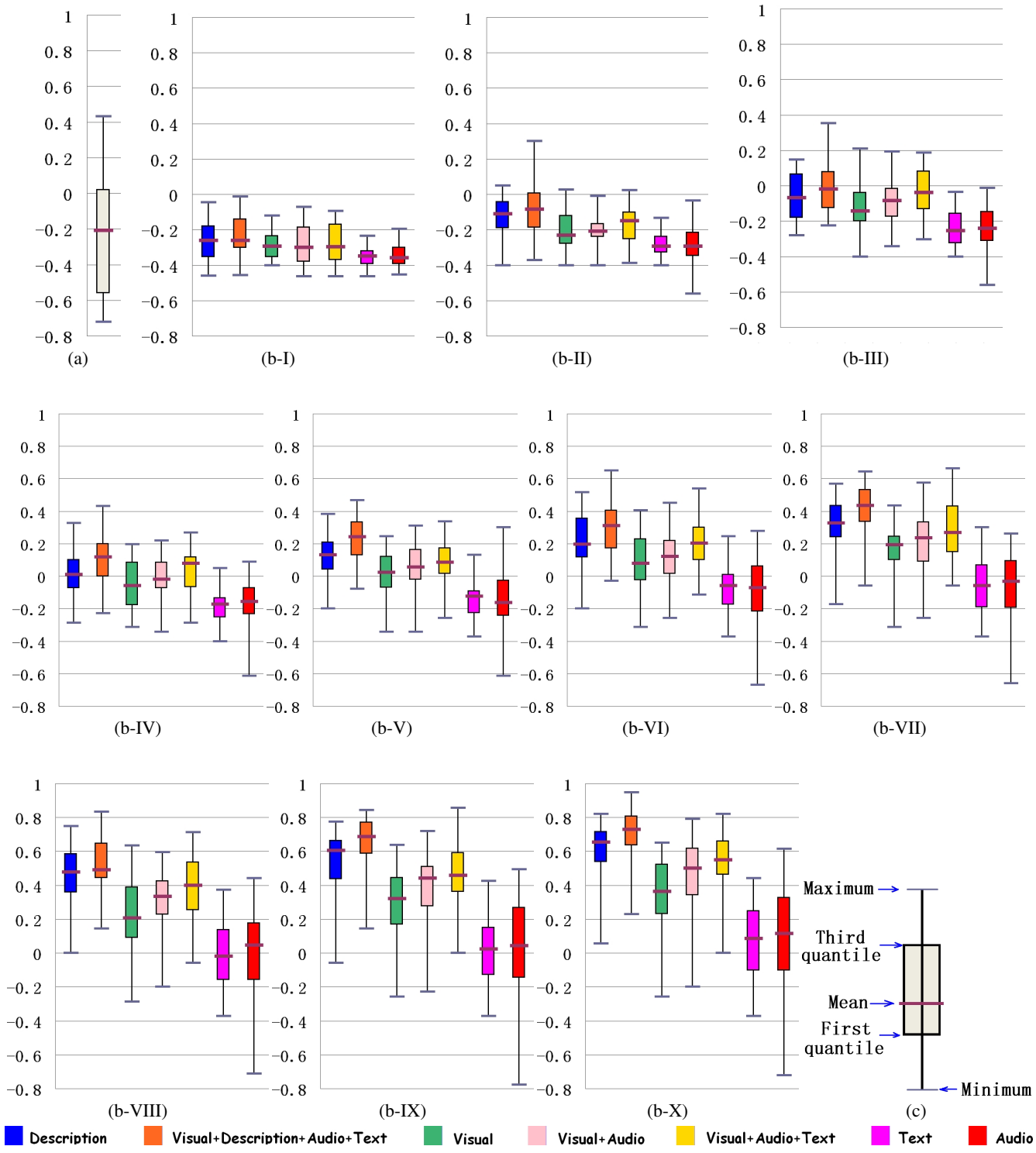

Figure 2: Box-plot diagram of the Kendall's Tau coefficients as measurement over the qualities of 20 personalized video ranking experiments using 20 different video search keywords. The Tau values of the initial YouTube video ranking is illustrated in (a). The box plots in (b-I)–(b-X) represent the Tau values with different similarity combinations after the users have watched the first one, two, and up to ten videos respectively. The statistic features indicated in a boxplot element are illustrated in (c). The corresponding numerical values of these performance statistics are also available in Table 1.

an online scheme. As mentioned earlier, the similarity measurement is very important for producing a quality personalized video ranking. During a typical search process, user feedback is usually available implicitly. Incorporating these additional clues into our personalized video ranking framework could further improve our system's overall performance.

7. ACKNOWLEDGEMENT

This work has a patent pending.

8. REFERENCES

[1] I. Ahmad, F. A. Cheikh, S. Kiranyaz, and M. Gabbouj. Audio-based queries for video retrieval over java enabled mobile devices. *Multimedia on Mobile Devices II*, 6074(1), 2006.

[2] A. Bernstein, E. Kaufmann, C. Kiefer, and C. Bürki. SimPack: A Generic Java Library for Similiarity Measures in Ontologies. Technical report, Department of Informatics, University of Zurich, 2005.

[3] S. A. Billings and G. L. Zheng. Radial basis function network configuration using genetic algorithms. *Neural Networks*, 8(6):877–890, 1995.

[4] P. Cano, E. Batlle, T. Kalker, and J. Haitsma. A review of audio fingerprinting. *Journal of VLSI Signal Processing Systems*, 41(3):271–284, 2005.

[5] S.-C. Cheung and A. Zakhor. Efficient video similarity measurement and search. In *ICIP '00: Proceedings of International Conference on Image Processing*, volume 1, pages 85–88, 2000.

[6] S.-C. Cheung and A. Zakhor. Estimation of web video multiplicity. In *Proceedings of SPIE Conference on Internet Imaging*, volume 3964, pages 34–46, 2000.

[7] S.-C. Cheung and A. Zakhor. Video similarity detection with video signature clustering. In *ICIP '01: Proceedings of International Conference on Image Processing*, volume 2, pages 649–652, 2001.

[8] S.-C. Cheung and A. Zakhor. Efficient video similarity measurement with video signature. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(1):59–74, 2003.

[9] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, second edition, 2001.

[10] C.-H. Hoi, W. Wang, and M. R. Lyu. A novel scheme for video similarity detection. In *CIVR '03: Proceedings of International Conference on Image and Video Retrieval*, pages 373–382, Urbana-Champaign, IL, USA, 2003.

[11] M. G. Kendall and J. D. Gibbons. *Rank Correlation Methods*. Oxford University Press, 1990.

[12] S. H. Kim and R.-H. Park. An efficient algorithm for video sequence matching using the modified hausdorff distance and the directed divergence. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(7):592–596, 2002.

[13] P. Knees, T. Pohle, M. Schedl, and G. Widmer. A music search engine built upon audio-based and web-based similarity measures. In *SIGIR '07: Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 447–454, New York, NY, USA, 2007. ACM.

[14] J. Lebosse, L. Brun, and J. C. Pailles. A robust audio fingerprint extraction algorithm. In *SPPR'07: Proceedings of IASTED International Conference on Signal Processing, Pattern Recognition, and Applications*, pages 269–274, Anaheim, CA, USA, 2007. ACTA Press.

[15] H. Li, D. Doermann, and O. Kia. Text extraction, enhancement and ocr in digital video. In *Lecture Notes in Computer Science*, volume 1655, pages 363–377. Springer, 1999.

[16] R. Lienhart and W. Effelsberg. Automatic text segmentation and text recognition for video indexing. *Multimedia Systems*, 8(1):69–81, 2000.

[17] T. Lin, C.-W. Ngo, H.-J. Zhang, and Q.-Y. Shi. Integrating color and spatial features for content-based video retrieval. In *ICIP '01: Proceedings of International Conference on Image Processing*, volume 3, pages 592–595, 2001.

[18] L. Liu, W. Lai, X.-S. Hua, and S.-Q. Yang. Video histogram: A novel video signature for efficient web video duplicate detection. Lecture Notes in Computer Science, 4352:94–103, 2006.

[19] B. Manjunath, J.-R. Ohm, V. Vasudevan, and A. Yamada. Color and texture descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):703–715, 2001.

[20] M. Park, H.-R. Kim, Y. M. Ro, and M. Kim. Frequency filtering for a highly robust audio fingerprinting scheme in a real-noise environment. *IEICE Transactions on Information and Systems*, E89-D(7):2324–2327, 2006.

[21] Y. Peng, C.-W. Ngo, C. Fang, X. Chen, and J. Xiao. Audio similarity measure by graph modeling and matching. In *MULTIMEDIA '06: Proceedings of ACM International Conference on Multimedia*, pages 603–606, New York, NY, USA, 2006. ACM.

[22] A. Rubinstein. *Lecture Notes in Microeconomic Theory*. Princeton University Press, 2006.

[23] B. Senechal, D. Pellerin, L. Besacier, I. Simand, and S. Bres. Audio, video and audio-visual signatures for short video clip detection: experiments on trecvid2003. In *ICME '05: Proceedings of IEEE International Conference on Multimedia and Expo*, 2005.

[24] J.-C. Shim, C. Dorai, and R. Bolle. Automatic text extraction from video for content-based annotation and retrieval. In *ICPR '98: Proceedings of International Conference on Pattern Recognition*, volume 1, pages 618–620, 1998.

[25] Y.-P. Tan, S. Kulkarni, and P. Ramadge. A framework for measuring video similarity and its application to video query by example. In *ICIP '99: Proceedings of International Conference on Image Processing*, volume 2, pages 106–110, 1999.

[26] E. Wold, T. Blum, D. Keislar, and J. Wheaton. Content-based classification, search, and retrieval of audio. *IEEE Multimedia*, 3(3):27–36, 1996.

[27] Y. Wu, Y. Zhuang, and Y. Pan. Content-based video similarity model. In *MULTIMEDIA '00: Proceedings of ACM International Conference on Multimedia*, pages 465–467, New York, NY, USA, 2000. ACM.

[28] S. Xu, H. Jiang, and F. C. Lau. Personalized online document, image and video recommendation via commodity eye-tracking. In *RecSys '08: Proceedings of ACM Conference on Recommender Systems*, pages 83–90, New York, NY, USA, 2008. ACM.

[29] P. Ziegler, C. Kiefer, C. Sturm, K. R. Dittrich, and A. Bernstein. Generic similarity detection in ontologies with the soqa-simpack toolkit. In *SIGMOD '06: Proceedings of ACM SIGMOD International Conference on Management of Data*, pages 751–753, New York, NY, USA, 2006. ACM.