Interactive Event Search Through Transfer Learning

Antony Lam,[1] Amit K. Roy-Chowdhury,[2] and Christian R. Shelton[1]

[1]Dept. of Computer Science & Engineering, University of California, Riverside
{antonylam,cshelton}@cs.ucr.edu
[2]Dept. of Electrical Engineering, University of California, Riverside
amitrc@ee.ucr.edu

Abstract. Activity videos are widespread on the Internet but current video search is limited to text tags due to limitations in recognition systems. One of the main reasons for this limitation is the wide variety of activities users could query. Thus codifying knowledge for all queries becomes problematic. Relevance Feedback (RF) is a retrieval framework that addresses this issue via interactive feedback with the user during the search session. An added benefit is that RF can also learn the subjective component of a user's search preferences. However for good retrieval performance, RF may require a large amount of user feedback for activity search. We address this issue by introducing Transfer Learning (TL) into RF. With TL, we can use auxiliary data from known classification problems different from the user's target query to decrease the needed amount of user feedback. We address key issues in integrating RF and TL and demonstrate improved performance on the challenging YouTube Action Dataset.*

1 Introduction

The growth of video sharing websites has resulted in a wealth of Internet videos (mostly of activities) available to users. Automated search of these videos present interesting challenges as the number of activities is arbitrarily large. In addition to the high variability of activities themselves, Internet videos typically exhibit greater variability in quality, camera movement, and lighting when compared with those of TV programs such as news broadcasts. Thus retrieval of such videos is still largely limited to the use of associated text tags.

However, search based on only text is limiting so direct analysis of video content is still desirable. The problem is that users could query for a vast array of activities and it would be very difficult to train high-level semantics for every possible query. In addition, if a user query were subjective (e.g. what the user thinks are "nice basketball shots"), there would be no way to train a system a priori for search. In this paper, we tackle these challenges in activity video retrieval through a combination of Relevance Feedback and Transfer Learning.

* This work was partially supported by NSF IIS 0712253 and the DARPA VIRAT program.

Fig. 1. Example of similarity between two different classes. If training data for "volleyball" were abundant while training data for "basketball" were scarce, the knowledge on classifying volleyball could be used to supplement the basketball training process.

To deal with difficulties in training systems for the vast array of queries users could make, Relevance Feedback (RF) [15] can be used and has been effectively applied to image retrieval [24]. The idea is to first search a database with respect to an initial query and return retrieval results to the user. If the user is dissatisfied with the results, user feedback on the relevance of retrieved items may be provided. The system could then use the feedback to better learn what the user has in mind and return refined results. If the user is still dissatisfied, then another iteration of user feedback may be repeated and retrieval results refined until the user is satisfied. Since user feedback is provided in RF, it is possible build custom classifiers in an online fashion for the user. Thus a wide range of queries can be made without the need to train them a priori.

However, a drawback of RF is when used to search videos of complex activities, a large amount of user feedback may be needed for good performance. (In other words, the few rounds of feedback a user would tolerate would provide too scarce a training set.) Transfer Learning (TL) [14] is a Machine Learning formulation where knowledge learned from one or more classification tasks is transfered over to a target task where the target task training data is scarce. If the abundant training data of source task(s) are *related* to the target task, it can be used to bias the classifier for the target task so that generalization performance can be improved.

As an example, consider the related activities "volleyball" and "basketball" (see Fig. 1). Say we are interested in classifying whether videos are of "basketball" but the amount of training data available is very limited. If the amount of training data for the task of classifying "volleyball" or "not volleyball" were abundant, the knowledge from the "volleyball" classification task could be used to supplement the training of the "basketball" task in order to improve generalized accuracy on classifying "basketball" videos.

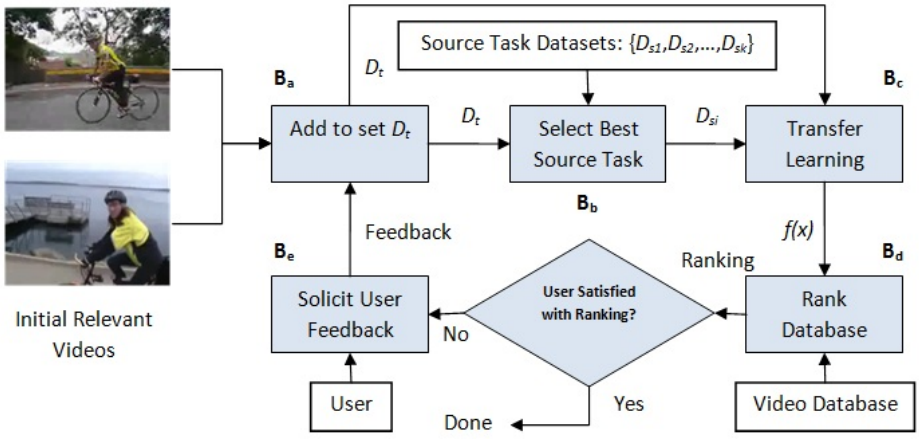

Fig. 2. Flowchart of system. Set D_t is initially empty before execution. After the first execution of block $\mathbf{B_a}$, set D_t should consist only of the initial relevant videos.

Provided system designers built a set of source task datasets for a variety of activities (this set of activities would only account for a small fraction of possible queries users could make), we could use the source data within a TL framework and combine it with RF to reduce the amount of needed user feedback. One of the key issues in combining RF and TL is determining which source task(s) are *related* to the target query, which is one of the main contributions of this work.

1.1 Overview and Contributions of Proposed Approach

Overview We now provide and overview of our proposed approach. In our formulation, the user first submits a few example videos representing their target query that can be used as initial queries to start the RF process. This is a reasonable assumption as it should be possible for users to obtain some sample videos at least *similar* to what they have in mind. For example, if a user wanted to find videos of cross country cycling, a few example videos of people riding bicycles in general should suffice. Such initial seed results might be obtained through a text query which often generates only a few relevant examples, especially when videos are only sparsely tagged. For example, a text search on Google.com for "rally racing video game" videos results in some relevant footage being retrieved but the search results are also swamped with footage from "X Games" rally races (real-life sporting events). If the user cannot refine his text query to improve search results, he can select the few relevant examples and use them to start a RF loop to refine his search results.

The basic flow of our proposed system is as follows: (The following steps are annotated with corresponding blocks in Fig. 2.)

1. Let D_t be an empty set.
2. User submits a few initial query examples of relevant videos.
3. The initial query examples are added to set D_t. (Block $\mathbf{B_a}$.)
4. The source task datasets and D_t are then processed in our algorithm (Sec. 3.3) for finding the best source task to transfer from. (Block $\mathbf{B_b}$.)
5. The best source task's training dataset and training data in D_t are then used in TL to obtain a classifier f for ranking the video database. (Block $\mathbf{B_c}$.)
6. The classifier f is used to rank the video database. (Block $\mathbf{B_d}$.)
7. The top N ranked videos from the database are then shown to the user.
8. If the user is satisfied with the results, the process terminates. Otherwise the system solicits user feedback (details to follow later). (Block $\mathbf{B_e}$.)
9. The user feedback is added to set D_t (block $\mathbf{B_a}$) and the process continues from step 4.

The feedback strategy in step 8 is a simple but effective approach to RF based on Active Learning [18]. Rather than solicit feedback on retrieved items, $\text{SVM}_{\text{Active}}$ [20] showed effective performance in image retrieval when soliciting feedback on the items considered most ambiguously relevant by the system.

Contribution Despite the effectiveness of $\text{SVM}_{\text{Active}}$ in image retrieval, the complexity of video activities limits the effectiveness of this framework. The main contribution of our work is in extending $\text{SVM}_{\text{Active}}$ to use TL for incorporating prior knowledge thus decreasing the required amount of user feedback. One of the key issues in combining RF with TL is in deciding what source task to transfer to the target task and we offer a solution in Sec. 3.3. As we explain in Sec. 2, our work is also one of the first to explore combining RF and TL. As we show in experiments on the YouTube Action Dataset [10], our framework provides benefits in improved ranking performance over standard RF frameworks for retrieval of complex activities.

2 Relation to Existing Work

Existing work in activity recognition demonstrates a trend of moving toward more complex activities. [8–10, 17] The main approaches of such work is to use new features, feature pruning techniques, and classification methods for improved complex activity recognition. However when applied to video retrieval, the subjectivity of human users is not modeled in these approaches. As a result, we propose a RF method that addresses this issue.

We note the main goal of this work is not to improve over previous work in terms of raw accuracy in activity recognition. Our focus is on the mechanism for quickly learning a user's subjective notions of activity class membership through user interaction. In fact, current work on designing features and algorithms for activity recognition is complementary to our work and could be integrated into our framework for overall improved retrieval. We now provide a review of related work in the two core tasks of our RF and TL framework for activity retrieval.

2.1 Relevance Feedback for Video Retrieval

In early work [11], RF for video was implemented by allowing the user to set weights in a scoring scheme utilizing various video features. In [4], this idea was extended to adaptively tune weights in a color and motion scoring scheme based on RF on top ranked videos. Other work [7] utilized more features such as speech recognition text, color, and motion in a weighted scoring scheme where weights were adaptively tuned based on RF of retrieved videos. In addition, semantic concept (e.g. "car") weightings were learned. A departure from the use weighted scores can be found in [13] where different scoring algorithms were adaptively chosen at each iteration of user feedback. However, the adaptive selection of scoring algorithms had to be manually trained by expert human users.

These systems showed good performance in their results. However, some can be complex, employing many different components. In applying them on larger scale problems, tuning the many parameters involved could be a daunting task.

Furthermore, most of the approaches described in this section do not make use of prior knowledge from the world to decrease the required amount of user feedback. While [7] used prior knowledge by explicitly building in high-level concepts like "cars", this approach requires learning a large number of classes that still would not cover the full range of queries users could make. We therefore address this issue by integrating TL into RF so that auxiliary training data of *different* classification problems from the target query can still be used to introduce prior knowledge into the system's learning process.

2.2 Basics of Transfer Learning

Before discussing related work in TL, we introduce a few TL concepts to provide context. In TL, there can be different relationships between the source and target tasks. Let task S be the source task and D_s be the source training set and task T be the target task and D_t be the target task's training set (where $|D_s| >> |D_t|$). Then TL can be subclassed into the following scenarios of interest:

1. S and T classify for the same class (e.g. running) but the distributions over the data for S and T are not the same. This is called the *Cross-Domain* problem in some work. As an example, if the training data D_s had been collected with camera A and D_t had been collected with camera B, simply combining D_s with D_t to improve classification accuracy on videos taken with camera B may not work well. (The cameras may have been positioned differently or have other differing characteristics.) The goal is to adapt the knowledge from D_s to augment the knowledge from D_t.
2. S and T classify for different but related classes. For example, S could be "volleyball" and T could be "basketball" (see Fig. 1). Since task S is related to task T, it should be possible to use the knowledge learned from D_s to improve generalization on D_t. This is the problem we focus on in this work.

There are more relationships between source and target tasks in TL described in [14] but the above mentioned ones are the most pertinent to our discussion.

2.3 Transfer Learning with Multiple Source Tasks

TL has been shown to be effective in transferring knowledge when source and target tasks are related. However, when there are multiple source tasks, deciding which to transfer from is still a difficult problem [14]. If a source task is too unrelated to the target task, transferring from such a source may result in *negative transfer* (transferring knowledge hurts target classification performance). The following work addresses TL in the presence of multiple source tasks.

In [23], the authors offer two methods for learning from multiple source datasets where some source tasks can be unrelated to the target. One method is effective but inefficient. The other finds a weighted linear combination of source classifiers and is efficient but only shows benefits when target data is very scarce.

In [21, 22], they propose the Adaptive-SVM (A-SVM) for regularizing a target Support Vector Machine (SVM) [1] hyperplane to be similar to a related source hyperplane while still fitting the scarce target training data. The problem they focus on is the *Cross-Domain* problem (see Sec. 2.2). For example, the detection of concepts such as "weather" between news programs on different TV stations. The editing style and camera work of different TV stations causes the data for the same classes to be distributed differently. In addition to transferring knowledge from related tasks, they also explore determining which source tasks would result in *positive transfer*. To achieve this, they determine which source classifiers have the best estimated performance on the target class. Since we use the $\text{SVM}_{\text{Active}}$ approach [20], the TL described in this work is most related to our focus. Thus we extend the ideas from [21, 22] beyond the Cross-Domain case.

Recent work related to A-SVMs [3, 6], present new mechanisms for Cross-Domain transfer of video actions and events. However, they do not present methods for source task selection. Furthermore, these mechanisms were designed for Cross-Domain transfer which may not be directly applicable to our problem of general TL. As the focus of the TL component in our work is in source task selection, we leave investigations into the possibility of adapting the transfer mechanisms in [3, 6] to general TL for future work. Finally, the related work mentioned here do not interact with the user which as mentioned before is crucial for capturing user subjective views of relevance.

2.4 Transfer Learning for Relevance Feedback Search

To the best of our knowledge there is no work on the general use of TL in RF. The RF surveys [5, 16, 24] do not even mention TL being applied to RF. Recent related work in the literature is mainly concerned with the Cross-Domain transfer problem for RF.

In [19], a study on how social tagged images could aid video search is presented. Their work is mainly concerned with how well manual relabeling of social tagged images without adaptation would work in a Cross-Domain scenario for video retrieval. They show results using RF and the benefits of simply cleaning up noisy labels without using adaptation. This framework does not apply in our case since we are working in a more general TL scenario.

In [12], two Cross-Domain learning methods are presented for RF. The first method uses a linear combination of the source and target classifier outputs with equal weighting. The second involves solving a regularized regression problem. Both methods performed similarly but combining the two via a heuristic for which method to use for each iteration of RF gave better overall performance.

While there is a little work on combining Cross-Domain transfer and RF in the literature, Cross-Domain transfer is only a special case of TL. The type of TL we explore involves transfer from different but *related* classification tasks and we offer a means of automatically determining task relatedness. Thus we present a complete system for RF search based on general TL. As stated earlier, this will also be one of the first explorations in combining RF and TL.

3 Relevance Feedback using Transfer Learning for Activity Search

3.1 Scoring Videos and Relevance Feedback

In this work, we assume that videos can be represented as fixed length vectors of extracted feature histograms such as STIP [9]. These vectors could then be used in SVM training of classification tasks. Once trained, the relevance *score* of a video is interpreted as its distance to the SVM decision surface where the higher the score, the more relevant a video. For example, if we used a linear SVM for scoring, we would have $score(\mathbf{x_i}) = \mathbf{w} \cdot \mathbf{x_i} + b$ where $\mathbf{w}$ is the normal to the SVM hyperplane, $\mathbf{x_i}$ is a video from the database, and b is the bias term.

Following the $\text{SVM}_{\text{Active}}$ framework [20], our system solicits feedback on the N videos the system finds most ambiguously relevant (those nearest the SVM hyperplane) and the user labels these videos as either relevant or irrelevant. Once relevance labels have been solicited from the user, the system can use the additional labels to retrain a more accurate classifier. This classifier could then be used to assign a new score to each video in the database and rerank them to better fit the user's target query. Our work extends $\text{SVM}_{\text{Active}}$ by incorporating TL. We now describe the components of our TL system.

3.2 Transferring Knowledge from a Source Task

Let D_s and D_t be the training data for source task S and target task T respectively. (Where $|D_s| >> |D_t|$.) Then ideally if the source and target tasks were the same, we could just train a more powerful classifier for the target task by augmenting D_t with D_s. In practice, the source and target tasks are unlikely to be the same but they could still be related. Then we could still augment D_t with D_s but with less weight given to the data in D_s.

We accomplish this by adjusting the C parameter in the SVM formulation. Recall that training an SVM involves solving the following optimization problem:

$$\min_{\mathbf{w},\xi}\{\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i\} \tag{1}$$

$$\text{s.t. } y_i(\mathbf{x_i} \cdot \mathbf{w} + b) - 1 + \xi_i \geq 0, \xi_i \geq 0$$

where $\mathbf{x_i}$ is the i^{th} datapoint and y_i, ξ_i are the label and slack variable associated with $\mathbf{x_i}$. $\mathbf{w}$ is the normal to the hyperplane. C is the parameter that trades off between training accuracy (high C) and margin size (low C).

Let D_{aug} be D_t augmented with D_s and let the data from D_s be indexed from 1 to n in D_{aug} while the data from D_t be indexed from $n+1$ to $n+m$ in D_{aug}. Then to weight the source data and target data in the SVM training of D_{aug} we solve the following:

$$\min_{\mathbf{w},\xi} \{\frac{1}{2}\|\mathbf{w}\|^2 + C_s \sum_{i=1}^{n} \xi_i + C_t \sum_{i=n+1}^{n+m} \xi_i\} \tag{2}$$

$$\text{s.t. } y_i(\mathbf{x_i} \cdot \mathbf{w} + b) - 1 + \xi_i \geq 0, \xi_i \geq 0$$

where all the variables are as described in Eq. 1 and C_s and C_t are the different parameters trading off the "hardness" versus "softness" of fitting the associated datapoint. (Note that we set $C_s < C_t$.)

We note that there was little difference between using all the source data to bias the target SVM and using just the support vectors from the source SVM. Since using only the support vectors results in faster training speeds, we train only on the source task support vectors in our implementation.

The A-SVM [21, 22] could have been used in place of this section's proposed method of transfer (which they call the "aggregate approach"). However the A-SVM does not offer benefits in improved accuracy over the aggregate approach and can even perform worse in some tests. The main advantage of using A-SVM is shortened training time. As the focus of this paper is on the feasibility of combining RF and TL for improved accuracy and the aggregate approach is more standard, we chose to use the aggregate approach.

3.3 Determining Which Source Task to Transfer From

Sec. 3.2 assumed we knew which source classifier to transfer from. However, transferring from the wrong classifier can hurt performance on the target task.

In [21], a number of strategies for choosing which source classifier to transfer from were presented. One method was to use score aggregation from multiple source classifiers. The basic idea was to use the "average" of multiple source classifiers with the hope that this would result in a more accurate classifier for assigning pseudo-labels to the unlabeled data. These pseudo-labels would then be used to evaluate how much individual source classifiers help improve ranking performance on the unlabeled examples. This approach does not work in our case. Since the authors were transferring knowledge in a Cross-Domain setting, all the source classifiers were assumed to classify for the same class. In our case, the source classifiers can be very unrelated to each other and thus combining an "average" of the source classifiers results in very poor performance.

Another proposed method was to assign scores to all unlabeled items using a potential source classifier (one trained on source data) and use the Expectation

Maximization (EM) algorithm to fit two Gaussian components to the scores. If the scores separate the data well then the means of the found Gaussian components should have greater distance between them. While a good idea, this is still not directly applicable to our problem because the target data are never used in this process; thus the same source classifier would always be selected regardless of the user feedback. However, if we first transfer the source classifier to the target classifier and then use the resulting classifier to score the unlabeled data, EM can be used to determine how well the transferred classification separates the data. We use this new procedure for determining which source classifier would help the target classifier produce the best separation of items in the database.

Formally, let D_s and D_t be the source and target training data and let $TL(D_s, D_t)$ be a function that produces a classifier where D_s was used to transfer knowledge to the target task (as described in Eq. 2). Then the following steps are taken to evaluate the quality of using D_s for the transfer:

1. Produce SVM $T_s = TL(D_s, D_t)$.
2. Use SVM T_s to compute scores (Sec. 3.1) Sc on the unlabeled database.
3. Use EM to fit Gaussian components $\mathcal{N}(\mu_1, \sigma_1^2)$ and $\mathcal{N}(\mu_2, \sigma_2^2)$ to scores Sc.
4. Determine the distance $d_\mu = (\mu_1 - \mu_2)^2$.

The distance d_μ can be used to indicate how well transferring the given source task to the target task would separate the unlabeled data (larger values are better). This provides an indication of whether the source task helps improve target task classification. The same procedure can be used to score the transfer for each of the available source tasks and the best source task could be chosen as the one to transfer from. We call this the **Score Clustering (SC)** method.

We note that projecting all source training data onto the subspace of the unlabeled database was found to be a helpful preprocessing step for determining what to transfer. Thus we first performed Principal Components Analysis on the unlabeled videos to obtain a set of basis vectors $\mathbf{V}$. We then projected *all* source task videos *and* unlabeled videos onto $\mathbf{V}$. So in our implementation, the projected videos were used instead of the original STIP histograms in all learning components of our system.

3.4 Integrating Relevance Feedback with Transfer Learning

We now formally describe the process of selecting a source task and transferring knowledge to the target task (user query) in the RF framework. Let $S_{Dtasks} = \{D_{s1}, D_{s2}, ..., D_{sk}\}$ be the set of source task training sets and D_t be the target task's training set. The TL portion of our framework operates as follows:

1. Given training data D_t from user feedback, determine the best source task training data D_{si} from set S_{Dtasks} to transfer from using SC (Sec. 3.3).
2. Use D_{si} to bias the learning of D_t using Eq. 2 and produce an SVM T_{si}.

SVM T_{si} is then used to rank the database of videos and if needed, feedback will be solicited on videos nearest the hyperplane of T_{si}. (Note that on each iteration of feedback, the choice of which task to transfer from is revisited.)

4 Experiments

Feature Representation and SVM Training We first converted all videos into fixed length vectors representing histograms of STIP features [9]. The first step to getting these histograms was to build a codebook of STIP features. We did so by taking 100,000 random STIP features from videos and using K-means to identify 1,000 centers. The set of centers were then treated as the codebook. Afterward, for each video in our experiments, we extracted its STIP features, quantized them according to the codebook, and created a 1,000 dimensional vector with counts how many occurrences of each type of quantized STIP feature was present in the video. For SVM training, we used the SVM and Kernel Methods Matlab Toolbox [2] and selected the linear kernel as it provided sufficient accuracy for our study.

Dataset We used the YouTube Action Dataset [10] in our experiments. This dataset consists of about 1,600 videos collected from YouTube.com with 11 categories of actions ranging from "basketball shooting" to "dog walking." Its videos are very challenging as they were taken outside of controlled settings and feature camera shake, differences in lighting, video quality, and camera position.

We note that in [10], their goal was to obtain *high classification accuracies* of video activities through new feature extraction and pruning techniques. Here, we are *not* attempting to obtain the best performance in terms of classification accuracies. Instead we are aiming to obtain the *best improvement* in performance through the use of TL. More sophisticated feature extraction and classification algorithms could be used in our framework but we chose to use standard features and learning algorithms so as to establish a control in our experiments.

Experimental Setup We chose all videos in the classes basketball, biking, diving, golf swing, and horse riding to be in our unlabeled database and all remaining videos to be source data. For TL, we set $C_s = 10^{-4}$ and $C_t = \infty$ in Eq. 2. There were a total of 778 videos in our unlabeled database with on average 150 videos per class. The source data was used to define a set of 1-versus-all classification problems (for example volleyball versus not volleyball). The target queries were for distinguishing one of the five classes listed above from the total unlabeled database. Feedback was seeded with five randomly selected positive and five randomly selected negative examples. Each query session involved three rounds of simulated user feedback where 10 examples nearest the SVM hyperplane would be labeled. By simulated, we mean that ground truth labels were used to judge the relevance of videos. In future work, we plan to compare system performance with simulated and real user feedback. Note that iteration 1 in the results only uses the initial examples from the user. Iteration 2 is when feedback is first used. Thus by iteration 4, the user would only have given feedback on $\mathbf{30/778} \approx \mathbf{4}\%$ of the database.

We also ran experiments on a variant of our system where no TL was used. That is, we replaced blocks $\mathbf{B_b}$ and $\mathbf{B_c}$ in Fig. 2 with a single block that only

takes in the target training data D_t and trains an SVM for it. In addition to testing against the no TL case, we also tested against a straightforward heuristic for source task selection. (To compare against our SC method.) If a source task S and target task T are related, we would expect TL from S to T to improve performance. Thus we did a set of experiments where $\mathbf{B_b}$ from Fig. 2 was replaced with the following procedure:

1. Given target task training data D_t, train an SVM T_t.
2. Determine the classification error of each source task training set D_{sj} with respect to SVM T_t.
3. Choose training set D_{si} with the lowest error as the source to transfer.

The intuition is if tasks S and T are related, using a classifier trained on one task's training data to classify the other should result in less degradation than if the tasks were not related. We call this the **Score Accuracy (SA)** method.

Metrics Used As different combinations of initial examples can affect performance, we tested querying for each category 100 times. (With the same initial queries used for both TL and non-TL tests.) We computed Average Precision (AP) to assess ranking performances (on only the currently unlabeled videos) for each iteration of feedback as:

$$AveragePrecision = \frac{1}{num}\sum_{r=1}^{N}(P(r) \times \mathrm{rel}(r)) \tag{3}$$

where $N = 50$ in our experiments, $P(r)$ is the precision at rank r, and $rel(r)$ is the indicator function for whether the r^{th} item in the ranking is relevant. We set $num = 50$ so AP values range from 0.0 to 1.0 with 1.0 being an ideal ranking.

A natural way to measure overall improvement from TL for all target queries would be to determine the average percent improvement in AP between corresponding TL versus no TL tests. However we found that although a majority of our tests resulted in positive transfer, there was a large amount of variation in percent improvement. For example, in one case we observed a AP value of 0.0016 for no TL but with TL, we obtained a AP of 0.4. In other cases, we observed improvements in AP of +0.2. So determining means and standard deviations in percent improvement does not adequately summarize our results.

Thus we plotted quartiles over *all* observed percent differences in our tests across the feedback iterations (Fig. 3) as this more adequately illustrates how our percent improvements in AP were distributed. The 50th percentile marks on the figure are the median percent improvements (as a function of feedback iterations) observed from all of the test runs conducted. The median line in the score clustering (SC) method's results indicates that half of all tests conducted resulted in at least about 20% improvement. The 25th percentile mark in the first iteration of the SC graph indicates that 75% of the tests resulted in some improvement from TL. Similarly, the first iteration 75th percentile mark in the SC graph shows that 25% of tests run resulted in over 100% improvement.

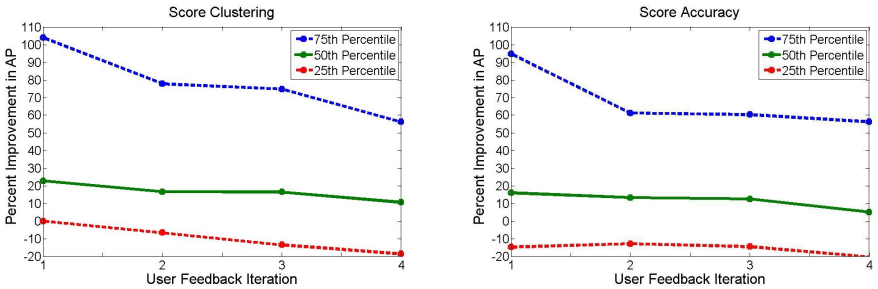

Fig. 3. Plots of percent improvement in AP of TL over not using TL for two methods of choosing source task: Score Clustering (left) and Score Accuracy (right). The distribution of improvements over all tasks and all initial video inputs are shown. Quartiles are plotted since percent improvements are highly varied but skewed toward positive values. The 50th percentile indicates the median percent improvement for a given iteration. The left graph's 75th percentile mark in iteration 1 indicates that 25% of the test queries had percent improvements over 100%. Note that iteration 1 only uses initial seeded examples from the user. Iteration 2 is where user feedback is first incorporated.

Results Fig. 3 indicates that SC is better than SA (see Sec. 4) in determining which source task to transfer from. This is probably because the SC method attempts to find which source task's bias would improve classification with respect to the target data on the particular unlabeled database being searched. So SC does not attempt to transfer knowledge for generalized performance and instead bases its criterion on the data being searched instead. The SA method does not consider any of the unlabeled data in the database which limits its ability to find a source task good for separating data on the database of interest. It is also not surprising that percent improvement tends to drop as the amount of user feedback is increased. As the amount of target task training data increases, one would expect the target classifier to generalize better without the need for TL.

While we could not show meaningful averages and standard deviations for individual percent improvements, we can show the overall Mean AP (MAP) for each class query to give readers a concrete idea of how MAP improves over feedback iterations. Results for TL (using SC for source task selection) and no TL are shown in Table 1. Fig. 4 also shows sample results for retrieval of "horse riding" videos for the first two user feedback iterations of the TL and no TL cases. (More such results are provided in the supplementary materials.)

5 Conclusion

We presented a framework in RF for complex activity video retrieval through a combination of RF and TL and demonstrated its utility on a real-life dataset of Internet videos. The primary contribution of this work was the use of EM to determine the best source task data to use for knowledge transfer resulting in overall less required user feedback in the search process. We also made one of the

Table 1. MAP for Different Queries (row) over Feedback Iterations (col.) The source tasks were soccer juggling, swing, tennis swing, trampoline jumping, volleyball spiking, and dog walking.

Transfer Learning				
Feedback Iteration	1	2	3	4
Basketball	0.26 ± 0.12	0.31 ± 0.14	0.34 ± 0.13	0.35 ± 0.12
Biking	0.55 ± 0.15	0.63 ± 0.14	0.70 ± 0.13	0.74 ± 0.13
Diving	0.21 ± 0.16	0.25 ± 0.15	0.29 ± 0.15	0.31 ± 0.15
Golf Swing	0.21 ± 0.12	0.26 ± 0.15	0.29 ± 0.17	0.26 ± 0.18
Horse Riding	0.19 ± 0.07	0.30 ± 0.11	0.40 ± 0.13	0.46 ± 0.13
No Transfer Learning				
Feedback Iteration	1	2	3	4
Basketball	0.17 ± 0.11	0.25 ± 0.13	0.28 ± 0.13	0.29 ± 0.13
Biking	0.49 ± 0.11	0.59 ± 0.12	0.66 ± 0.12	0.72 ± 0.11
Diving	0.13 ± 0.13	0.18 ± 0.15	0.24 ± 0.15	0.28 ± 0.15
Golf Swing	0.14 ± 0.12	0.16 ± 0.14	0.19 ± 0.15	0.23 ± 0.16
Horse Riding	0.21 ± 0.09	0.28 ± 0.12	0.34 ± 0.15	0.41 ± 0.16

first explorations of combining RF with general TL. As the key problem in this framework is the choice of source task data to transfer, we hope to improve on our current results in the future through improvements in source task selection.

References

1. Burges, C.: A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery **2** (1998) 121–167
2. Canu, S., Grandvalet, Y., Guigue, V., Rakotomamonjy, A.: SVM and kernel methods matlab toolbox. Perception Systmes et Information, INSA de Rouen, Rouen, France (2005)
3. Cao, L., Liu, Z., Huang, T.: Cross dataset action detection. In: CVPR, IEEE (2010)
4. Chen, L., Chin, K., Liao, H.: An integrated approach to video retrieval. In: ADC, Australian Computer Society, Inc. (2008)
5. Crucianu, M., Ferecatu, M., Boujemaa, N.: Relevance feedback for image retrieval: a short survey. State of the art in audiovisual content-based retrieval, information universal access and interaction including data models and languages, DELOS2 Report (FP6 NoE) (2004)
6. Duan, L., Xu, D., Tsang, I., Luo, J.: Visual event recognition in videos by learning from web data. In: CVPR, IEEE (2010)
7. Hauptmann, A., Lin, W., Yan, R., Yang, J., Chen, M.: Extreme video retrieval: joint maximization of human and computer performance. In: MULTIMEDIA, ACM (2006)
8. Hu, Y., Cao, L., Lv, F., Yan, S., Gong, Y., Huang, T.: Action detection in complex scenes with spatial and temporal ambiguities. In: ICCV, IEEE (2009)
9. Laptev, I.: On space-time interest points. International Journal of Computer Vision **64** (2005) 107–123

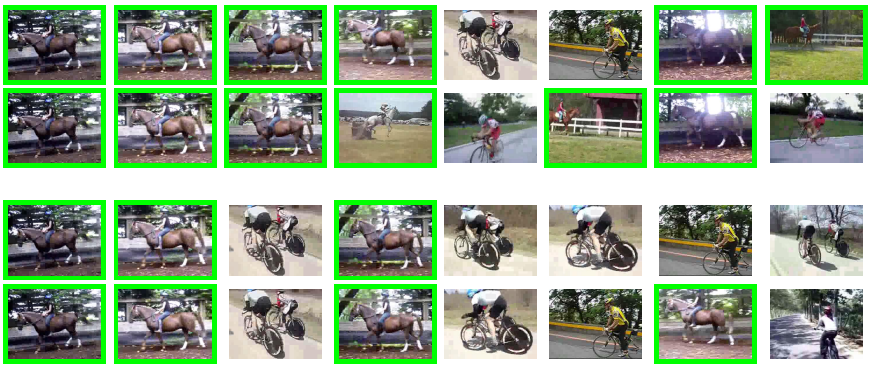

Fig. 4. Ranking results for "Horse Riding." The first two rows show the top 8 videos for the first and second feedback iterations with TL. The bottom two rows are the first and second feedback iterations without TL. With TL, there is less confusion between biking and horse riding *and* a greater variety of relevant videos are captured.

10. Liu, J., Luo, J., Shah, M.: Recognizing realistic actions from videos "in the wild". In: CVPR, IEEE (2009)
11. Liu, X., Zhuang, Y., Pan, Y.: A new approach to retrieve video by example video clip. In: MULTIMEDIA, ACM (1999)
12. Liu, Y., Xu, D., Tsang, I., Luo, J.: Using large-scale web data to facilitate textual query based retrieval of consumer photos. In: MULTIMEDIA, ACM (2009)
13. Luan, H., Zheng, Y., Neo, S., Zhang, Y., Lin, S., Chua, T.: Adaptive multiple feedback strategies for interactive video search. In: CIVR, ACM (2008)
14. Pan, S., Yang, Q.: A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering (2009)
15. Rocchio, J. In: Relevance Feedback in Information Retrieval. Prentice-Hall, Inc. (1971) 313–323
16. Ruthven, I., Lalmas, M.: A survey on the use of relevance feedback for information access systems. Knowledge and Engineering Review **18** (2003) 95–145
17. Ryoo, M., Aggarwal, J.: Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In: ICCV, IEEE (2009)
18. Settles, B.: Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison (2010)
19. Setz, A., Snoek, C.: Can social tagged images aid concept-based video search? In: ICME, IEEE (2009)
20. Tong, S., Chang, E.: Support vector machine active learning for image retrieval. In: MULTIMEDIA, ACM (2001)
21. Yang, J., Yan, R., Hauptmann, A.: Cross-domain video concept detection using adaptive SVMs. In: MULTIMEDIA, ACM (2007)
22. Yang, J., Hauptmann, A.: A framework for classifier adaptation and its applications in concept detection. In: MIR, ACM (2008)
23. Yao, Y., Doretto, G.: Boosting for transfer learning with multiple sources. In: CVPR, IEEE (2010)
24. Zhou, X., Huang, T.: Relevance feedback in image retrieval: A comprehensive review. Multimedia Systems **8** (2003) 536–544