Active Inference for Retrieval in Camera Networks

Daozheng Chen[1], Mustafa Bilgic[2], Lise Getoor[1], David Jacobs[1], Lilyana Mihalkova[1], and Tom Yeh[1]
[1]Department of Computer Science, University of Maryland, College Park, MD 20742
[2]Department of Computer Science, Illinois Institute of Technology, Chicago, IL 60616
[1]{dchen, getoor, djacobs, lily, tomyeh} @cs.umd.edu
[2]mbilgic@iit.edu

Abstract

We address the problem of searching camera network videos to retrieve frames containing specified individuals. We show the benefit of utilizing a learned probabilistic model that captures dependencies among the cameras. In addition, we develop an active inference framework that can request human input at inference time, directing human attention to the portions of the videos whose correct annotation would provide the biggest performance improvements. Our primary contribution is to show that by mapping video frames in a camera network onto a graphical model, we can apply collective classification and active inference algorithms to significantly increase the performance of the retrieval system, while minimizing the number of human annotations required.

1. Introduction

Camera networks frequently contain hundreds if not thousands of cameras, generating huge amounts of data. Querying such networks to find relevant frames quickly is therefore a daunting task. Determining whether a lost child has left a mall, or finding where a terror suspect entered the subway may require human operators to comb through large volumes of video. Computer vision algorithms, including those for person reidentification, aim to automate this process. However, it is difficult to achieve sufficient performance for critical tasks using fully automatic methods. In addition, the majority of these methods do not explicitly reason about the structure of the camera network during identification.

In this paper, we show how we can discover and exploit spatial and temporal relationships among frames in a camera network, and we study the use of active inference, which can be used to direct human labeling efforts to portions of video whose labels will provide the biggest performance improvements. We consider a frame from a camera network to be "relevant" if it contains a queried person; the retrieval task is to identify all of the relevant frames.

We perform this by first mapping video frames in a camera network onto a graphical model. This allows us to perform more effective inference than when each frame is independently analyzed. More importantly, as a human operator examines frames of the video, her input can improve classification of portions of video that have not yet been analyzed. In addition, we can use active inference algorithms to direct attention towards the most useful portions of video. Our primary contribution is therefore to show that by modeling video frames using a graphical model, we can perform collective classification and active inference to produce more effective video analysis in camera networks.

Specifically, our contributions are as follows:

- We describe how to model retrieval in a camera network using a graphical model.
- We develop active inference techniques to prioritize frames for human annotation.
- We empirically show that, among the several active inference approaches we consider, the technique that most heavily exploits network structure gives the best performance.

The paper is organized as follows. We formulate the problem in Section 2. We describe two probabilistic frameworks for video analysis in camera networks in Section 3. Next, we describe the active inference techniques in Section 4. Section 5 discusses the experimental setup and results. We discuss related work in Section 6, and conclude in Section 7.

2. Problem Formulation

Let $\mathcal{C}$ denote a network of cameras and let $\mathcal{F}_C$ represent the set of frames taken by camera $C \in \mathcal{C}$.

Each frame $F \in \mathcal{F}_C$ is represented by a feature vector $\vec{X}_F = \langle X_F^1, X_F^2, \ldots, X_F^p \rangle$ and class label Y_F pair, $F = \langle \vec{X}_F, Y_F \rangle$. Here, the $\vec{X}_F$ are continuous variables; these variables can depend on the specific query that is being processed, and indicate the similarity between the query image and a video frame (described further in Section 5.3). Each Y_F is binary, indicating whether F is *relevant* or *irrelevant* to a query.

Given training data $\mathcal{D}^{tr} = \{\langle \vec{X}_F^{tr}, Y_F^{tr} \rangle\}$ for $F \in \mathcal{F}_C, C \in \mathcal{C}$, a test set $\mathcal{D}^{te} = \{\langle \vec{X}_F^{te}, Y_F^{te} \rangle\}$, and a budget B determining the number of labels a human annotator can provide, our objective is to determine the best set of labels $\mathcal{A} \subseteq \mathcal{Y}^{te}$ to acquire as follows:

$$\underset{\mathcal{A} \subseteq \mathcal{Y}^{te}, |\mathcal{A}| \leq B}{\operatorname{argmax}} \quad Reward(S^{te} \mid \mathcal{X}^{te}, \mathcal{Y}^{te}, \mathcal{A})$$

where $\mathcal{Y}^{tr}$ and $\mathcal{Y}^{te}$ represent the set of labels for the frames from the training and testing data respectively, $S^{te} = \{L_1, L_2, \cdots, L_N\}$ is the set of random variables for the label of each of the N testing instance, and $\mathcal{X}^{tr}$ and $\mathcal{X}^{te}$ have similar meaning for sets of features. In practice, this reward function is based on the conditional probabilities of the labels given observed, acquired and inferred information. We use a probabilistic model to estimate these probabilities. We consider two types of $Reward$ functions in this paper; the first one is accuracy, measuring the percentage of frames in $\mathcal{Y}^{te}$ that are correctly classified. The second one is average precision, measuring how well the model can rank the frames in order of relevance.

3. Probabilistic Models

We will contrast two probabilistic models for video retrieval in a camera network.

3.1. Local Models (LM)

In the simplest case, we assume that, given the parameters of the underlying model, the labels of all frames in the network are independent of one another, given the features of the frame. Thus, in this model, we assume that each Y_F depends only on $\vec{X}_F$ and nothing else. Because this model uses only the local information about the current frame, we call it a *local model* (LM).

For estimating $P(Y_F \mid \vec{X}_F)$, any probabilistic classifier that can be trained discriminatively, such as logistic regression, can be used. In our experiments, we use a visual bag-of-words model [22], which has been shown useful for video image retrieval. The query image and video frames are represented as vectors of visual word frequencies. We then compute cosine similarity between these frequency vectors to represent $\vec{X}_F$, which is then used as input to the probabilistic classifier. We provide more details in Section 5.3.

3.2. Relational Models (RM)

Because one person is typically present or not present for a duration of time in a camera, and because cameras have overlapping and non-overlapping fields of view, we expect the above independence assumption to miss important relationships in the data. So we also consider a *relational model* (RM), where the information from *neighboring* frames is integrated. Specifically, to predict the label Y_F, we use the label information from three types of neighbors, which we define below.

1. **Temporal neighbors** $\mathcal{N}_{Y_F}^{T^k}$**:** These are the labels of the frames that appear k time steps before frame F and k time steps after it in the same camera C.

2. **Positively correlated spatial neighbors** $\mathcal{N}_{Y_F}^{P}$**:** These are the labels of the frames from other cameras at the same time step that tend to have the same label as F. Such neighbors may correspond to cameras with overlapping fields of view and can be discovered from the training data.

3. **Negatively correlated spatial neighbors** $\mathcal{N}_{Y_F}^{N}$**:** These are the labels of the frames from other cameras at the same time step that tend to have labels different from Y_F. For example, when cameras have non-overlapping fields of view, a person can be present in at most one camera. Such neighbors can also be discovered automatically.

The set of neighbors of Y_F is then defined as $\mathcal{N}_{Y_F} = \mathcal{N}_{Y_F}^{T^k} \cup \mathcal{N}_{Y_F}^{P} \cup \mathcal{N}_{Y_F}^{N}$. Relational models use both $\vec{X}_F$ and $\mathcal{N}_{Y_F}$ to predict Y_F. However, because the neighbor labels are also often unobserved, the labels in the test data need to be inferred collectively. *Collective classification* is the process of using a relational model to infer the labels in a network simultaneously, exploiting the relationships between the network entities (see [20] for an overview). In this paper, we use Iterative Classification Algorithm (ICA). We describe it below.

ICA uses two models, a local model and relational model, to infer the labels of related entities iteratively. It learns a local model that uses only $\vec{X}_F$ to bootstrap the labels, and then applies a relational model that uses both $\vec{X}_F$ and $\mathcal{N}_{Y_F}$ to propagate the labels to neighboring frames in the network iteratively. Specifically, the relational model component of ICA represents each frame F as a vector that is a combination of $\vec{X}_F$ and features that are constructed using $\mathcal{N}_{Y_F}$.

Because frames from different cameras can have varying numbers of neighbors, the combined feature vector $\langle \vec{X}_F, \mathcal{N}_{Y_F} \rangle$ will be of different length for different frames. To get a fixed-length vector representation, we use an aggregation function aggr over the neighbor labels. For example, count aggregation constructs a fixed-size feature vector by counting the number of neighbors with each label. With this aggregation, we build the following combined feature vector: $\vec{X'}_F = \langle \vec{X}_F, \mathtt{aggr}(\mathcal{N}^{T^k}_{Y_F}), \mathtt{aggr}(\mathcal{N}^{P}_{Y_F}), \mathtt{aggr}(\mathcal{N}^{N}_{Y_F})\rangle$. Once the features are constructed, then an off-the-shelf probabilistic classifier can be used to learn $P(Y_F \mid \vec{X'}_F)$. Despite its simplicity, ICA has been shown to be quite effective and efficient [14, 17].

3.2.1 Choosing Neighborhoods for Cameras

In this paper, we use the explicit time information in each camera to define the temporal neighborhood. Let $F^t_{C_i}$ represent the frame from camera C_i at time step t. Then,

$$\mathcal{N}^{T^k}_{Y_{F^t_{C_i}}} = \{Y_{F^{t-k}_{C_i}}, Y_{F^{t-k-1}_{C_i}}, \ldots, Y_{F^{t-1}_{C_i}}, Y_{F^{t+1}_{C_i}}, Y_{F^{t+2}_{C_i}}, \ldots, Y_{F^{t+k}_{C_i}}\}$$

We learn the positive-spatial and negative-spatial neighborhoods from the data as follows. Let $\mathcal{Y}_{C_i}$ represent the temporally ordered set of all frames from camera C_i in the testing data. Then,

$$\mathcal{N}^{P}_{Y_{F^t_{C_i}}} = \{Y_{F^t_{C_j}} \mid corr(\mathcal{Y}_{C_i}, \mathcal{Y}_{C_j}) > \sigma_p\}$$

and

$$\mathcal{N}^{N}_{Y_{F^t_{C_i}}} = \{Y_{F^t_{C_j}} \mid corr(\mathcal{Y}_{C_i}, \mathcal{Y}_{C_j}) < \sigma_n\}$$

where $corr(.,.)$ measures the correlation between two ordered sets, and σ_p and σ_n are threshold parameters that define whether a camera should be included as a neighbor.

4. Active Inference

We allow the underlying retrieval algorithm to request the correct labels for some frames at inference time. This setup is called "active inference" meaning that the underlying model can actively collect more information at inference time [18]. The goal is to make the most of available human resources. We would like to determine for which frames the probabilistic model should request labels so it can label the remaining frames as well as possible. In this section, we describe a general framework for active inference and several possible algorithms for video analysis.

We considered the following active inference techniques:

1. **Random sampling (RND):** Sample frames randomly across different cameras and time steps.

2. **Uniform sampling (UNI):** Sample frames uniformly over time, for each camera.

3. **Most relevant (MR):** Sample frames whose probability of being relevant is the highest, where the probability is based on the output of the probabilistic model.

4. **Uncertainty sampling (UNC):** Sample the frames whose entropy value is the highest, where the entropy is defined using the probability estimates of the probabilistic model.

5. **Most likely incorrect (MLI):** Sample the frames that are most likely to be incorrectly predicted. For this, we adapt the reflect-and-correct algorithm (RAC) [1], which uses a separately trained classifier to predict which instances in a general network classification problem are likely to be misclassified. Below we describe how we adapt RAC for the purposes of retrieval in a camera network.

The first four methods can be applied to both LM and RM, and our experiments demonstrate that RM with relational information outperforms LM significantly. Because MLI is based on RAC, which specifically targets collective classification, it can be applied with only RM.

4.1. Adapting RAC for Retrieval in Camera Networks

RAC is an active inference technique that works as follows. At training time, a separate classifier is trained to predict whether or not an instance is misclassified. Then, at inference time, RAC uses this classifier to predict at what instances RM is likely to make a mistake, and suggests acquiring the label of a central instance in a region where most of the instances are predicted to be misclassified [1]. To learn and predict whether an instance is misclassified, RAC utilizes a few features that are indicative of misclassification. At a higher level, these features are based on the RM prediction, LM prediction, and some global statistics about the data. RAC learns the classification function using the trainind data used for training RM.

In this paper, we introduce two important modifications of the original RAC framework. These address i) what features to use for RAC in camera networks and ii) how to train RAC. To distinguish this adapted version from the original, we refer to our version as *Most Likely Incorrect* (MLI).

4.1.1 Features for MLI

We used the following 10 features as possible indicators of misclassification:

1. Four features based on the probability estimates of RM. We use the entropy of the probability estimate for the single label Y_F and average entropy values over $\mathcal{N}^{T^k}_{Y_F}$, $\mathcal{N}^{P}_{Y_F}$, and $\mathcal{N}^{N}_{Y_F}$. These features capture the uncertainty of the frame and uncertainty of its neighborhood.

2. Four features based on the probability estimates of RM and LM. We use the KL divergence between the

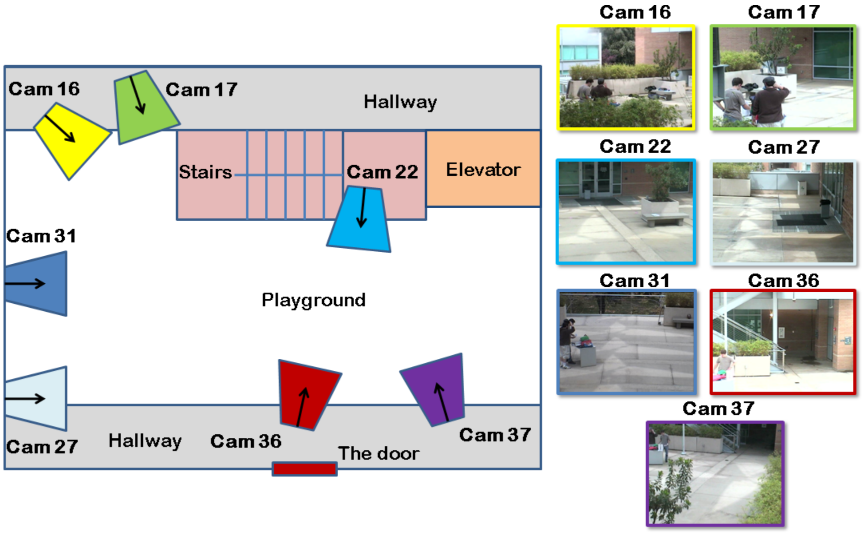

Figure 1. Camera layout and sample frames from each of the 7 cameras. The camera ID above each frame is the actual ID used in the UCR Videoweb dataset.

RM probability and LM probability for label Y_F, and the average of this value for $\mathcal{N}_{Y_F}^{T^k}$, $\mathcal{N}_{Y_F}^{P}$, and $\mathcal{N}_{Y_F}^{N}$. These features provide a way to measure the likelihood a frame and its neighborhood are misclassified, since disagreement between RM and LM is a sign of misclassification.

3. Whether Y_F is predicted to be relevant by RM. This feature captures whether there is any bias of the model toward one class. This feature is expected to be especially useful for domains where there is a class skew in the data.

4. Percentage of $\mathcal{N}_{Y_F}^{P}$ and $\mathcal{N}_{Y_F}^{T^k}$ that agree with the label Y_F. $\mathcal{N}_{Y_F}^{P}$ and $\mathcal{N}_{Y_F}^{T^k}$ both have positive correlation with Y_F. A lower percentage value indicates higher likelihood of misclassification.

4.1.2 Training MLI

Because MLI predicts whether a frame is misclassified, it cannot use the labels in the training data directly. Rather, it needs to be trained on data that specifies the features described above for each frame and whether the frame is misclassified. To construct this training data, we split the original training data into k folds. We train RM and LM on $k-1$ folds and test them on the k^{th} fold. For each training frame F for MLI, we construct the features described above using these RM and LM. The label of F for MLI is *misclassified* if RM (trained on the $k-1$ folds) predicts Y_F incorrectly and *not-misclassified* otherwise. We repeat this process for each fold.

5. Experimental Evaluation

We performed video retrieval on the Videoweb dataset from UC Riverside [4], where various people are recorded for short periods of time, called the *scenes*. Our video retrieval task is: given training data for a number of people in a number of scenes, retrieve the frames for a new query person (whose image is given) in a new scene. We train our probabilistic models, LM and RM, on the training data, and perform active inference on the test data, where a human annotator can provide the labels of a small number of frames, and the probabilistic models are expected to utilize the annotated frames to perform better on the remaining frames. We next describe the dataset, constructing the local features from the query image and video frames, our evaluation strategy, and experimental results in detail.

5.1. Dataset

The dataset contains a number of scenes recorded over four days. Each scene is recorded by a camera network and the videos from different cameras in the network are approximately synchronized and contain several types of activities over a number of people.

We arbitrarily choose scenes 20 to 25 for our experiments. In these scenes, seven cameras overlook the playground. Scene 21 does not include data from one of the seven cameras, so we use it to generate queries. All other scenes are used for retrieval. The time period for scene 24 is approximately twice as long as those of other scenes. We split it into two parts with equal time periods, and refer to them as scene 24.1 and 24.2. This gives us six scenes of approximately equal length. Each camera has about half an hour of video over all scenes, and we use a frame rate of one frame per second. Figure 1 shows the camera layout and example frames from these seven cameras.

5.2. Queries

We use a set of query images from four persons. These images are from scene 21, which is not included in the scenes used for retrieval. We consider three query images for each person from the front, back, and side view. Since people in the dataset can easily be occluded and are mainly characterized by the patterns of their clothes, we manually crop each query image to highlight their distinctive clothing regions. These cropped images are used as queries. Figure 2 shows the query images and their cropped results.

5.3. Similarity Features

Both LM and RM need the local feature vector $\vec{X}_F$ for each frame, query and scene. We adopt a commonly used, bag-of-words [22] approach to derive a feature that measures the similarity between the query and regions of interest in each frame. This involves computing image descriptors at keypoints in a region of interest. These descriptors are quantized into codewords, which are created by applying k-means clustering to training examples. Then histograms of the codewords in two regions of interest are compared using cosine similarity. In our implementation, the entire query is one region of interest, while we use the background subtraction algorithm of Zivkovic [31], which is based on a standard method using Gaussian mixture model [24], to determine regions of interest in the video. We densely sample keypoints in the region of interest, and build descriptors using a color histogram over RGB space. For each video frame, descriptors from all detected regions of interest are considered as a whole to represent the frame. In preliminary experiments, the color histogram is more effective than some other descriptors, such as SIFT [11] and OpponentSIFT [26]. Figure 3 shows an example of densely sampled key points from video frames. Using k-means clustering on a random subset of descriptors, we form 500 visual words. By comparing histograms, we obtain features that encode the similarity between a query and video frames.

5.4. Training LM, RM, and MLI

When testing for a particular query in a given scene, the neighborhood structure of the camera network, the probabilistic models LM and RM, and the active inference technique MLI are learned on data from other persons and other scenes. For computing the temporal neighborhood, $\mathcal{N}_{Y_F}^{T^k}$, we set $k = 1$ for RM, and $k = 4$ for MLI. We have three queries for each person, and they all share the same structure, probabilistic models, and MLI. The threshold σ_p for learning positive-spatial neighbors is 0.6 and σ_n for negative-spatial neighbors is -0.15. We use logistic regression in WEKA with default parameters [7] to learn LM, RM, and MLI. We generated the ground truth for each person by manually labeling the frames. Figure 4 shows an example of the learned network structure.

5.5. Non-incremental and Incremental Sampling

The sampling locations for RND and UNI do not depend on the output of any probabilistic models. Thus, for them sampling is carried out independently, in a non-incremental fashion, and the locations sampled for a smaller budget are not a subset of those sampled for a larger budget. On the other hand, sampling for MR, UNC, and MLI is dependent on the output of probabilistic models. Because RM inference is based on the acquired labels, the labels acquired at lower budget levels can change the predictions of RM. Thus, for these active inference techniques we perform incremental sampling, first acquiring the labels of a small subset of k frames, then incorporating these acquired labels into the prediction, and running the acquisition algorithm again to sample the next set of k frames. We do this until the budget is used up. In our experiments we used $k = 10$.

5.6. Evaluation Methods

We perform active inference for both LM and RM with a budget (the number of frames for which the human annotator provides the labels during inference) varying from 0% to 50% of all frames. For UNC-RM, MLI, and MR-RM we repeatedly perform inference to update the predictions whenever ten new labels have been acquired. In these methods, the use of inference can allow the results of partial labeling to guide the system in determining locations for additional labeling.

Given that we have six scenes, four people, and three queries per person, we train on five scenes with nine queries, and test on a scene for three queries, and we repeat this process for each scene and each person, giving us 72 different test cases. We trim the scenes so that each one is 270 seconds (4.5 minutes).

For each active inference technique, we plot two performance measures on the Y axis as a function of the budget on the X axis. The first performance measure is accuracy, measuring the percentage of frames predicted correctly. The second measure is the 11-point average precision [16] of the precision-recall (PR) curves over all frames. For those frames whose labels are acquired, we set their probabilities to either 0 or 1 based on their actual labels. However, in three out of 72 cases, the queries are completely absent from the scene and the PR curve is undefined for these three cases. We ignore these three scenes for calculating the 11-pt precision measure. Significance claims are based on a paired t-test at the 0.05 level.

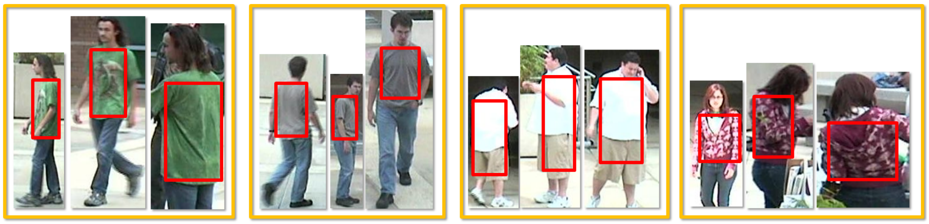

Figure 2. The 12 query images from 4 people. The parts inside the red bounding boxes are the cropped portions used to compute similarity measure.

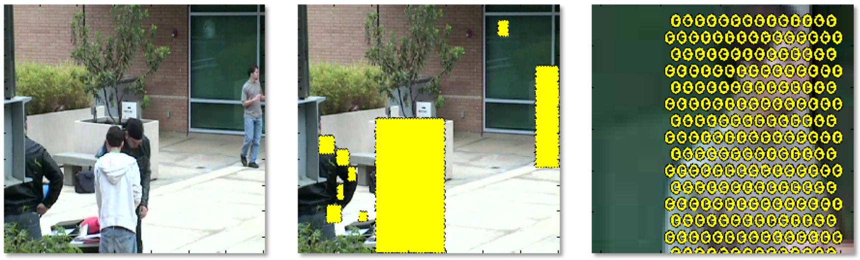

Figure 3. An example showing densely sampled points over regions of interest computed by background subtraction. The right-most figure is the enlarged view of the left-most region where key points are densely sampled. The detected region is of square shape, because we run morphological operations after background subtraction and extract non-overlapping bounding boxes over connected components. In addition, the reason that the person with a black coat is only partially sampled is because he has been present over a long period of time with little motion.

5.7. Results and Discussion

We compare the performance of four active inference methods described in section 4 using LM and RM, while considering MLI only with RM. For MLI, we use temporal neighbors within four time steps for constructing the features that are based on temporal neighbors. The left side of Figure 5 shows performance using average accuracy, while the right side shows 11-pt average precision. For LM, UNC has the best performance when compared with RND, UNI, and MR. Therefore, we show results for only UNC for LM in order to increase readability in the graphs. For RM, however, we show the results for all active inference techniques, as they are better than UNC using LM.

Based on a statistical analysis of the results, we draw the following conclusions. First, whenever we apply the same algorithm using LM and RM, RM performs significantly better. Comparing UNC-LM and UNC-RM in Figure 5 provides a typical example of the large magnitude of this difference. Second, we find that UNC-RM and MLI always perform significantly better than all other methods. Third, MLI has a statistically significant advantage over UNC-RM in terms of accuracy up to 32% budget (600 labels), and the two methods are comparable afterwards. When we measure 11-pt average precision, MLI is significantly better than UNC-RM in a few spots, and never significantly worse. Based on this result, we conclude that the use of graphical models and collective classification provides large improvements in performance for active inference. In addition, MLI, our adaptation of RAC, provides the best performance, especially at low budget levels, which are more likely to be used in practice.

6. Related Work

Person reidentification, in which a person seen in one surveillance video is located in later ones, closely resembles the query problem we address. Wang et al. [30] use shape and appearance context to model the spatial relationships of object parts to do appearance-based person reidentification. Gray and Tao [6] design a feature space and use Adaboost to learn person reidentification classifiers. Lin and Davis [10] reidentify people by a pairwise comparison-based learning and classification approach. Loy et al. [13] facilitate human reidentification by using cross canonical correlation analysis to learn the spatial temporal relationship of video frames in a camera network. In contrast, local descriptors have been widely used in object recognition. In particular, Sivic and Zisserman [22] consider video retrieval using a bag-of-words model.

Other work has used graphical models to represent camera networks. Loy et al. [12] performs abnormal event detection by modeling regions from different camera views using a time delayed probabilistic graphical model. Chen et al. [3] use a conditional random field (CRF) to model a camera network identify a known set of people.

Tracking over camera networks has also been widely ad-

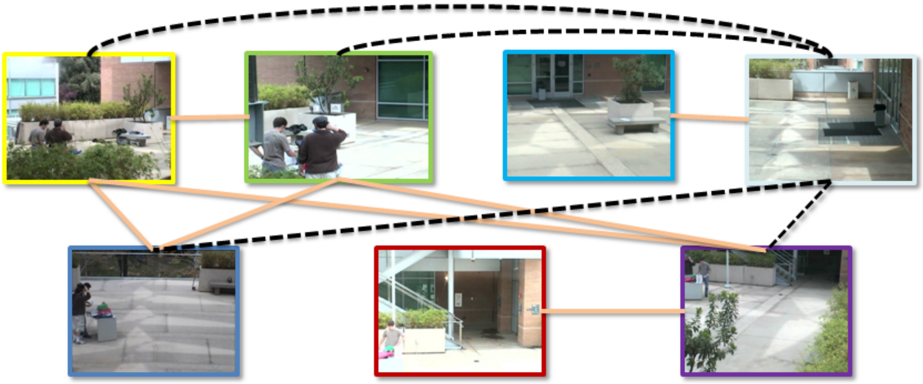

Figure 4. An example of learned topology. Light gold edges with solid lines denote positive correlation and black edges with dashes denote negative correlation. Temporal edges are not shown because they are fixed.

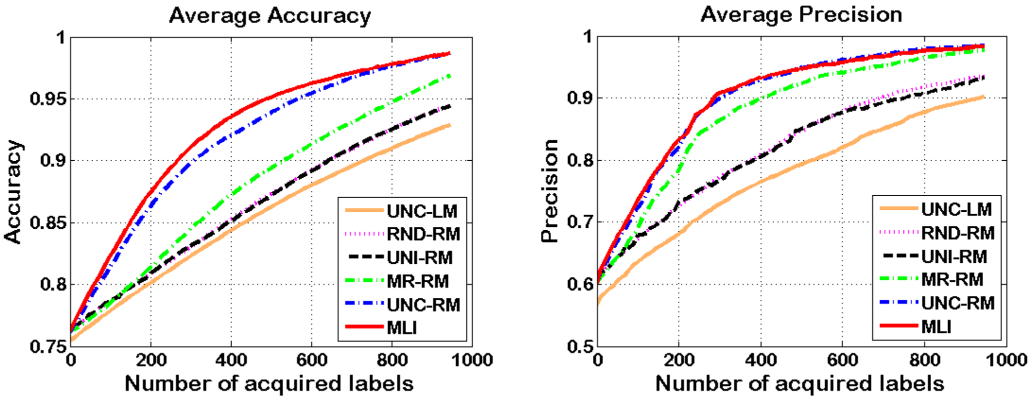

Figure 5. **Left**: Average accuracy as budget increases. **Right**: Average precision as budget increases.

dressed (eg., [15, 25, 5, 8, 23]). Typical problems include inferring the topology of the camera network [15, 25, 5] and finding correspondences between trajectories from multiple cameras [8, 23].

The key difference between our work and these is the use of collective classification and active inference to handle queries to a camera network. In a very different context, some of these issues have been addressed in interactive segmentation. For example, Rother [19] extends the graph-cut method [2] with substantially simplified user interaction to achieve superior image segmentation quality. Wang et al. [29] interactively extract foreground objects from a video using user painted strokes. While this work focuses on minimizing the need for human labeling over still images or a single video, we focus on active inference methods that can direct human attention over camera networks.

Krause and Guestrin [9] did a theoretical analysis of active inference for graphical models and they showed that the optimal solution is tractable for Hidden Markov Models, but it is $\mathbf{NP}^{\mathbf{PP}}$-hard for graphical models even with a polytree structure. Rattigan et al. [18] performed active inference on networks of arbitrary structure by first grouping the nodes of the network into clusters and then acquiring the labels of the central nodes in the clusters. Finally, the active learning work [21] is very related to active inference, and it has been applied to visual recognition [27, 28]; however, the biggest difference is that active learning acquires labels to construct training data to learn a model, whereas active inference performs label acquisition for an already learned model to guide the probabilistic inference to achieve better accuracy and precision.

7. Conclusion

Our work addresses the problem of using active inference to direct human attention in searching a camera network for people that match a query image. We first use local information to measure the similarity between the query and each frame. We find that by representing the camera network using a graphical model, we can more accurately determine whether video frames match the query, and improve our ability to direct human attention. We experiment with five methods of determining which frames should be labeled. We find that the value of the graphical model is very strong, regardless of which algorithm is used to select frames for human labeling. In comparing these active inference methods, we find that there is an advantage in labeling those frames that are most likely to contain errors. This can be captured by a simple method that measures the entropy of the probability distribution that indicates our uncertainty about the labels of each frame. However, we find that we do somewhat better by adapting an approach that uses a classifier to predict which frames are in error. Overall, we demonstrate that we can adapt tools developed for active inference in graphical models to improve the capacity of humans to effectively search or annotate video from camera networks.

Acknowledgement: This work was supported by the Army Research Office, ARO #W911NF0810466. Lilyana Mihalkova is supported by a CI Fellowship under NSF Grant #0937060 to the CRA.

References

[1] M. Bilgic and L. Getoor. Reflect and correct: A misclassification prediction approach to active inference. *ACM TKDD*, 3(4):1–32, 2009. 3

[2] Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. In *ICCV*, 2001. 7

[3] D. Chen, A. Bharusha, and H. Wactlar. People identification through ambient camera networks. In *ICDE Worshop on Multimedia Ambient Intelligence, Media and Sensing*, 2007. 6

[4] C. Ding, A. Kamal, G. Denina, H. Nguyen, A. Ivers, B. Varda, C. Ravishankar, B. Bhanu, and A. Roy-Chowdhury. Videoweb Activities Dataset, ICPR contest on Semantic Description of Human Activities (SDHA). http://cvrc.ece.utexas.edu/SDHA2010/Wide_Area_Activity.html, 2010. 4

[5] R. Farrell, D. Doermann, and L. Davis. Learning higher-order transition models in medium-scale camera networks. In *ICCV workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras*, 2007. 7

[6] D. Gray and H. Tao. Viewpoint invariant pedestrian recognition with an ensemble of localized features. In *ECCV*, 2008. 6

[7] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witten. The weka data mining software: An update. *SIGKDD Explorations*, 11, 2009. 5

[8] O. Javed, K. Shafique, and M. Shah. Appearance modeling for tracking in multiple non-overlapping cameras. In *CVPR*, 2005. 7

[9] A. Krause and C. Guestrin. Optimal nonmyopic value of information in graphical models - efficient algorithms and theoretical limits. In *IJCAI*, 2005. 7

[10] Z. Lin and L. Davis. Learning pairwise dissimilarity profiles for appearance recognition in visual surveillance. In *ISVC*, 2008. 6

[11] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 5

[12] C. Loy, T. Xiang, and S. Gong. Modelling activity global temporal dependencies using time delayed probabilistic graphical model. In *ICCV*, 2009. 6

[13] C. Loy, T. Xiang, and S. Gong. Multi-camera activity correlation analysis. In *CVPR*, 2009. 6

[14] Q. Lu and L. Getoor. Link based classification. In *ICML*, 2003. 3

[15] D. Makris, T. Ellis, and J. Black. Bridging the gaps between cameras. In *CVPR*, 2004. 7

[16] C. Manning, P. Raghavan, and H. Schtze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008. 5

[17] J. Neville and D. Jensen. Iterative classification in relational data. In *AAAI Workshop on Learning Statististical Models from Relational Data*, 2000. 3

[18] M. Rattigan, M. Maier, and D. Jensen. Exploiting network structure for active inference in collective classification. In *ICDM Workshop on Mining Graphs and Complex Structures*, 2007. 3, 7

[19] C. Rother, V. Kolmogorov, and A. Blake. ”grabcut”: interactive foreground extraction using iterated graph cuts. *ACM SIGGRAPH*, 23(3):309–314, 2004. 7

[20] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93–106, 2008. 2

[21] B. Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009. 7

[22] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *ICCV*, 2003. 2, 5, 6

[23] B. Song and A. Roy-Chowdhury. Stochastic adaptive tracking in a camera network. In *ICCV*, 2007. 7

[24] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. *CVPR*, 1999. 5

[25] K. Tieu, G. Dalley, and W. Grimson. Inference of non-overlapping camera network topology by measuring statistical dependence. In *ICCV*, 2005. 7

[26] K. van de Sande, T. Gevers, and C. Snoek. Evaluating color descriptors for object and scene recognition. *IEEE TPAMI*, 32(9):1582–1596, 2010. 5

[27] S. Vijayanarasimhan and K. Grauman. Whats it going to cost you?: Predicting effort vs. informativeness for multi-label image annotations. In *CVPR*, 2009. 7

[28] S. Vijayanarasimhan, P. Jain, and K. Grauman. Far-sighted active learning on a budget for image and video recognition. In *CVPR*, 2010. 7

[29] J. Wang, P. Bhat, R. A. Colburn, M. Agrawala, and M. F. Cohen. Interactive video cutout. *ACM SIGGRAPH*, 24(3):585–594, 2005. 7

[30] X. Wang, G. Doretto, T. Sebastian, J. Rittscher, and P. Tu. Shape and appearance context modeling. In *ICCV*, 2007. 6

[31] Z. Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *ICPR*, 2004. 5