What to Reuse?: A Probabilistic Model to Transfer User Annotations in a Surveillance Video

Omar Florez[†], Curtis Dyreson[*], Junaith Shahabdeen[†],
[*]Utah State University
Email: Curtis.Dyreson@usu.edu
[†]Intel Labs
Email: {Omar.Florez,Junaith.Ahemed.Shahabdeen}@intel.com

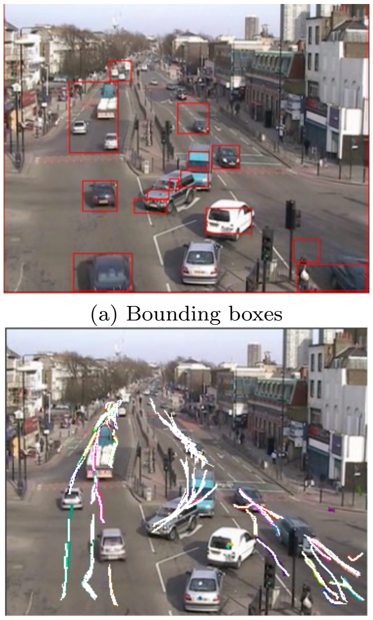
***Abstract*—Techniques to extract or understand interactions between moving objects in video is becoming increasingly important as the amount of video increases. Applications in surveillance range from understanding traffic to studying fish schooling behavior. Because of the massive amount of data, fast, approximate techniques based on statistical models are common. These models connect user annotations (*labels*) to scenes in a (short) video segment. The connection forms a *domain*, which associates information about moving objects in scenes with the labels, such as to indicate whether a user considers a particular traffic scene to be "dangerous." Unfortunately a statistical model trained in one domain often yields low precision and recall when applied to another domain because the random variables that explain video content exhibit changing marginal and conditional probability distributions over time (e.g., due to different backgrounds, changes in illumination, shading, and numbers of moving objects). This problem is exacerbated when new domains continuously arise (e.g., in the real-time processing of video) and user annotations are only limited to training data, a common scenario for surveillance video.**

In this paper, we propose a new, cross-domain technique that reuses labeled content from source domains to improve the prediction of user annotations in a target domain. Our model probabilistically learns how users annotate scenes based on the similarity of target to source domains. Two domains that are similar will share a large number of observable features. We encode the similarity in a covariance matrix, which flexibly allows allows users to set an arbitrary covariance structure between pairs of domains before training the model.

Experiments show that our method improves state-of-the-art techniques (SVM and CF) in predicting dangerous scenes in real-world traffic surveillance videos.

I. Introduction

Surveillance cameras are widely-used to monitor and alert responders to potentially interesting events, e.g., an oil spill in a pipeline, or an accident in a busy intersection. Since near real-time response to interesting events is desirable in these applications, the video produced by the cameras must be processed in real-time. Let's assume that the potential responders annotate a small portion of a video with labels about events of interest, e.g., a car runs a red light. Ideally, a video processing system could then predict from this small training set of labels when interesting events occur later in the video. In other words, the system should reuse the annotations to improve prediction in future scenes. The problem of predicting future labels in video has three aspects that go beyond traditional machine learning algorithms.



(a) Bounding boxes

(b) Trajectories

Fig. 1. Describing surveillance video for traffic roads. (a) After moving objects have been segmented, bounding boxes enclose activity information (b) Bounding boxes are tracked during a time window to extract trajectories, highlighted in white.

1) Mixed membership — The mixture components that form a video scene are not known in advance; rather they should be learned based on the frequent co-occurrence of discrete activities in the video. A generative process, like Latent Dirichlet Allocation (LDA) [1], is needed to generate a proportion of topics for every scene.
2) Dynamic content — Video is noisy, with lots of moving objects that cannot be isolated cleanly. Hence, activities in a video cannot be extracted consistently. As LDA generates topics based on

activity co-occurrence, it yields different types of topics at various places in the video. This decreases the utility of applying a statistical model trained in one part of the video to another.

3) User annotations are scarce and expensive — We can ask users to provide labels for some scenes via crowd-sourcing, but realistically, the user annotations will be limited to a small portion of the video. Repeating this process when every new external variable happens is not scalable in a continuously growing video database. Somehow the model has to learn how to reuse the limited set of labels to best fit the remaining video.

To address these aspects, we propose a new model that we call the Crossdomain Probabilistic Model (CPM). The basic idea behind CPM is to generate topics that jointly model activities of two domains. Those topics become the common latent variables to propagate labels from a source to a target domain. The similarity of two domains, in terms of the number of common activities, guides the inference of latent variables.

This paper is organized as follows. Section II gives definitions and common terminology. Related work is discussed in Section III. Section IV studies methods to connect content in different domains. Section V introduces our algorithm. Section VI shows experiments that demonstrate the usefulness of the proposed approach for real traffic video. Finally, Section VII concludes the paper.

II. Background

The problem of understanding video involves three kinds of information: *features*, *activities*, and *scenes*.[1] A *feature* is a trajectory that exhibits the temporal behavior of a moving object, for example, consider the trajectories of vehicles going up, down, and turning right in Figure 1 (c). An *activity* groups trajectories with similar shape. One way to compute activities is to hash each trajectory, those that end up in the same bucket form an activity. CPM uses the Timeseries Sensitive Hashing (TSH) algorithm proposed in [4]. TSH can map similar trajectories to the same bucket, even when those trajectories vary in length, with high probability. Figure 2 illustrates the process of finding activities. In the figure, trajectories are hashed into four buckets. The hashing yields a dictionary of discrete activities (e.g., A, B, C, and D). A *scene* is a time window containing some number of activities.

To this standard set of terms we add *annotation* and *domain*. An *annotation* is a user description of a scene, e.g., a user could label a scene as "safe" or rate the amount of violence in a scene as "high." A *domain* is a collection of user annotations.

Below we formally define each of the terms described informally in the previous paragraphs.

- An *activity*, a_i, is the basic unit of discrete data. It is an entry in a fixed dictionary of V terms.
- A *topic*, β_k, is a distribution over a subset of activities.
- A *scene*, m, is a time window containing a collection of N activities denoted by $A = \{a_1, a_2, ..., a_N\}$. We can characterize a scene, m, as a proportion, θ_m, over K existing topics.
- An *annotation* is a numeric value $r = \{0, 1\}$ provided by a user to represent his/her perception of a scene (e.g., *is this scene dangerous?*). Without loss of generality, we discuss only a simple, two-valued space, which could be extended to the interval [0-1].
- A *domain*, D, is a matrix of users, I, as rows and scenes, M, as columns. Users annotate the scenes of a domain with the value r_{im},

$$r_{im} = \begin{cases} 1, & \text{if scene } m \text{ is annotated} \\ 0, & \text{otherwise} \end{cases}$$

.

As explained before different domains will generate the content of video scenes in different ways in response to external variables. This makes it difficult to predict the labels of scenes in new domains.

III. Related Work

A. Latent Dirichlet Allocation (LDA)

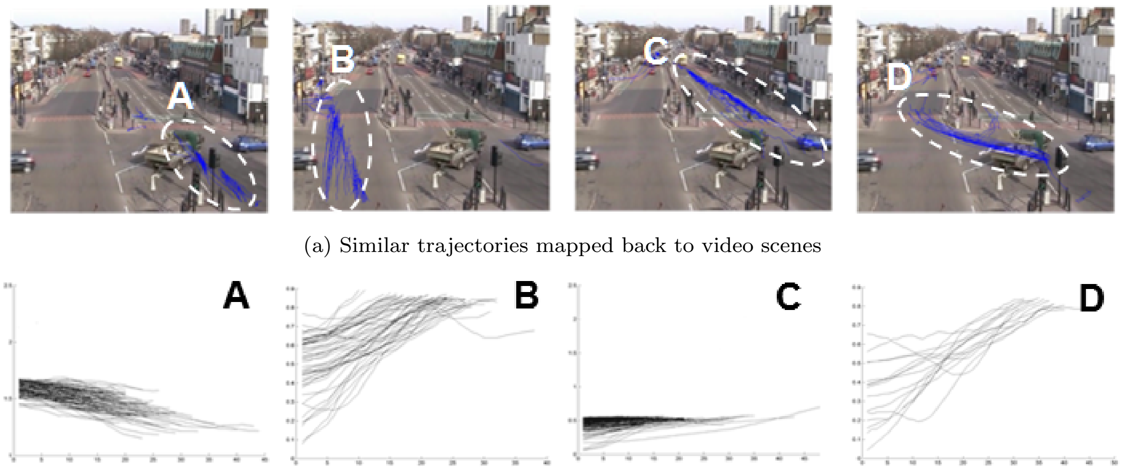
We use LDA to generate topics in video. A topic is a distribution over activities that co-occur frequently in scenes. Every scene is thus formed by a multinomial proportion over K topics.

Let's fix the following parameters of the model: K topics, β (each β_k is a vector of probabilities over the V entries of the dictionary of activities), and the Dirichlet parameter α (a vector of K components with $\alpha_i > 0$). LDA models every scene, m, with the following generative process:

1) Choose a topic proportion θ_m from the distribution Dirichlet(α)
2) For each of the N activities,
   a) Choose topic id z_{mn} from the multinomial distribution Mult(θ_m)
   b) Choose an activity a_{mn} from the multinomial distribution Mult($\beta_{z_{mn}}$)

The above process explains how LDA allows a scene to exhibit a mixed membership over the K possible topics. For example, LDA can capture that the scene shown in Figure 1 (c) contains topics β_1 and β_2, where β_2 corresponds to a collection of two frequently co-occurring activities (vehicles going down and vehicles turning right) in the second lane. A latent variable for this scene represents the proportions over $K = 5$ possible topics showing that topics β_1 and β_2 are active, $\theta = \{0.6, 0.4, 0.0, 0.0, 0.0\}$. Note that similar scenes will exhibit similar topic proportions, θ. Hence, θ_m provides a low-dimensional representation for the content of a video scene, m.

[1] This terminology is used in the Computer Vision community.



(a) Similar trajectories mapped back to video scenes

(b) Content of some buckets in the hash table

Fig. 2. Discovery of activities in video. (a) Similar trajectories show a similar shape in video scenes. (b) They are mapped to buckets A, B, C, and D of TSH.

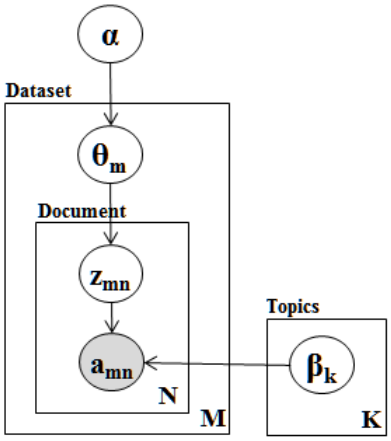


Fig. 3. Graphical model representation of LDA. Activities are shadowed to represent an observable variable.

LDA is formed of conditional relationships between activities, topics, and parameters. Figure 3 shows these dependencies as a probabilistic graphical model. It includes *hidden variables* (per-activity topic assignments, z_{mn}, per-scene topic proportions, θ_m, and per-dataset topic distribution, β_k), *observable variables*, activities, a, and a *dataset-level parameter* (Dirichlet parameter α). The hidden variables reflect a space of latent variables in the video, so its posterior inference is needed to estimate which topics best fit to the generation of activities in each scene. The posterior distribution over all the random variables in the model can be decomposed into a product of conditional distributions[2], as follows.

$$p(\theta, z|a, \alpha, \beta) \propto \prod_{n=1}^{N} p(\theta|\alpha) \prod_{n=1}^{N} p(z_n|\theta) p(a_n|z_n, \beta) \qquad (1)$$

We can use variational Expectation Maximization (EM) as a deterministic alternative to Markov Chain Monte Carlo (MCMC) to approximate the computation of the posterior distribution $p(\theta_{1:M}, z_{1:M}|a_{1:N}, \alpha_{1:K}, \beta_{1:K})$ as in [1]. This will optimize independently the variational parameters that govern the latent variables of Equation (1) by minimizing the Kulllback-Leibler (KL) divergence between the variational distribution and the true posterior $p(\theta, z|w, \alpha, \beta)$.

B. Activity Mining

The recognition of activities in video is an open problem that has recently received much attention. Commonly, low-level visual features and actions have been modeled and classified to provide interpretation of activities. While the traditional way to categorize existing research is by motion representation such as local features (e.g., changes in velocity, motion trajectories, and gradients) or global features (e.g., key frames), recent research has employed hierarchical Bayesian models such as LDA [1] and Hierarchical Dirichlet Process (HDP) [5] to cluster those activities into scenes [6], [7], [8]. The above research has led to techniques that can discover atomic scenes, but such techniques omit the complex interactions between scenes commonly present in video. Wang et al. [7] approach this problem by adding one more level to the hierarchy of the LDA and HDP Bayesian models and providing extended versions of integral probabilistic hierarchical Bayesian models (LDA, HDP, and Dual-HDP mixture models) to cluster moving pixels into atomic activities and interactions. Similarly, Li et at. [9] infer global behavior patterns through modeling behavior correlations using a hierarchical probabilistic Latent Semantic Analysis (pLSA). Both techniques, however,

[2] Both Equation (1) and Figure 3 are equivalent as they represent conditional dependence between random variables

temporal variations in the distribution of activities when they learn global interactions. In contrast, our technique models video scenes with topics that do not assume the same probability of co-occurring activities over time, a reasonable scenario imposed by the processing of continuous video streams.

IV. Domain Adaptation

The mining of activities across multiple domains is a problem that exhibits new characteristics. Existing algorithms model scenes as a combination of topics in a single domain [7], [6]. Though topics from one domain are partially preserved when projected to other domains, the source and target domains could have different vocabularies of activities (marginal distributions) and the dependency between latent variables (conditional distributions) may vary. These issues change the fundamental assumptions of topic modeling for mining activities in video.

Observe that the content a domain is transferable to a new domain if both domains share a set of common variables (which can be observable or latent). Our goal is to discover domain-independent variables that connect similar content in two domains. We compare the transference at three levels: feature-level, topic-level, and cross-domain level to determine which the best level.

A. At a feature-level

A simple approach is to find a set of observations (activities) present in the dictionary of both domains, as seen in Figure 4 (a) and then perform a K-means clustering over these common features to generate K mixture components. Then, in each scene we obtain a low-dimensional representation on the content by computing a histogram of activities indexed by each of the K clusters and normalizing.

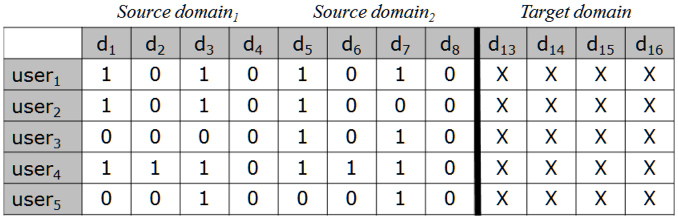
B. At a topic-level

We compute topics with LDA in each domain independently and define a similarity measure between topics in terms of the number of activities that they share across domains. Every topic is connected to its most similar topic in the other domain, as seen in Figure 4(b). When we separate connected topics, we end up with a new set of mixture components that connects content in both domains. As before, we compute the content for each scene by counting the number of activities indexed by each mixture component.

C. At a crossdomain-level

The previous two approaches are sub-optimal solutions to find latent variables that are relevant for domain adaptation. This is because either video activity is noisy and unstable (feature-level approach) or domain-specific topics model co-occurrence of activities in each domain, but fail to represent mutual co-occurrence across domains (topic-level approach).

We improve the above two methods by collapsing activities of both domains in a combined dataset and extracting topics to have a common latent variable to explain both domains. By considering two video segments as a single domain, we can learn topics that model activities in two domains jointly. That is, a collection of topics becomes a common latent variable that shares co-occurrence information across two domains, as shown in Figure 4(c). We take special care to consider the same number of activities in each domain to avoid bias in the generation of topics for a particular domain. Our experiments show that this method provides better recall values than feature- and topic-level techniques.



	Source domain₁				*Source domain₂*				*Target domain*			
	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_{13}	d_{14}	d_{15}	d_{16}
$user_1$	1	0	1	0	1	0	1	0	X	X	X	X
$user_2$	1	0	1	0	1	0	0	0	X	X	X	X
$user_3$	0	0	0	0	1	0	1	0	X	X	X	X
$user_4$	1	1	1	0	1	1	1	0	X	X	X	X
$user_5$	0	0	1	0	0	0	1	0	X	X	X	X

Fig. 5. Multiple source and target domains, missing annotations are shown with `X`s.

V. Crossdomain Probabilistic Model

Consider three source domains and a current target domain as depicted in Figure 5. A scene, m, in the source domains is annotated as dangerous ($r_{im} = 1$) or non-dangerous ($r_{im} = 0$) by a user, i. Scenes in the target domain are not annotated ($r_{im} =\mathtt{X}$). We need a model that predicts those missing annotations based on the similarity of the content in the source and target domains and the similarity of the user annotations.

A source domain can be factorized into latent vectors $u_i \in \mathbb{R}^K$ (for users) and $v_m \in \mathbb{R}^K$ (for scenes), respectively. The annotation of user i to scene m in the original matrix can be approximated as an inner product between their corresponding latent vectors,

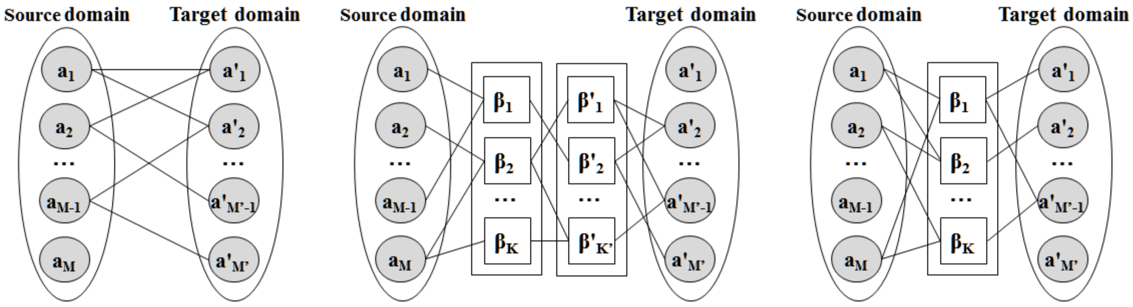
$$r_{im} \sim u_i^T v_m \tag{2}$$

which are learned by minimizing the least squared error with respect to the original user annotations r_{im},

$$min_{U,V} \sum_{i,m} (r_{im} - u_i^T v_m)^2 + \lambda_u \left\| u_i \right\|^2 + \lambda_v \left\| v_m \right\|^2 \tag{3}$$

with regularization parameters λ_u and λ_v.

Note that the inner product $u_i^T v_m$ corresponds to a prediction of whether user i considers the interaction of activities contained in scene m as dangerous. However, such a prediction cannot be computed with high precision in a *target domain* because scenes in the target domain do not contain user annotations and therefore r_{im} is not



(a) Feature-level (b) Topic-level (c) Cross-domain level

Fig. 4. Three different strategies for Domain Adaptation. The random variables $a_{1:M}$ and $\beta_{1:K}$ represent activities and latent variables, respectively.

available for Equation 3, as visually described in the target domain of Figure 5.

We introduce the Crossdomain Probabilistic Model (CPM) to alleviate that problem. CPM learns user annotations on latent variables that connect the generation of content across a pair of source and target domains (see Section IV-C and Figure 4(c)). CPM uses those cross-domain topic proportions, θ_m, in place of the latent vector v_m of Equation (3) to factorize the user annotations in both domains, as follows.

$$r_{i,m} \sim \mathcal{N}(u_i^T \theta_m, c_{ij}^{-1}) \tag{4}$$

c_{im}^{-1} being a precision parameter.

However, such predictions should include an uncertain quantity ϵ_m that offsets the topic proportion in response to the dissimilarity of their corresponding domains. This information is encoded using a normal distribution with expected value equal to the number of activities in the target domain and variance equal to the number of different entries in the dictionary of activities of both domains.[3] The idea is to penalize source domains that are different in terms of feature representations. Thus, a latent vector v_m explains a scene m as $v_m = \theta_m + \epsilon_m$, where $\epsilon_m \sim \mathcal{N}(\mu, \Sigma)$, is equivalent to

$$v_m \ \sim \ \mathcal{N}(\theta_m + \mu, \Sigma)$$

which models when the document latent vector v_m is close to its topic-proportions θ_m. This makes a user annotation equivalent to,

$$\begin{aligned} \mathrm{E}[r_{i,m}|u_i, \theta_m, \epsilon_m] \ &\sim \ \mathcal{N}(u_i^T(\theta_m + \mu), \Sigma)) \\ &\sim \ \mathcal{N}(u_i^T v_m, \Sigma) \end{aligned} \tag{5}$$

This cross-domain factorization can also be generated as a probabilistic graphical model with the following generative process,

1) For each of the I users,
   a) Choose a latent vector u_i from $\mathcal{N}(0, \lambda_u^{-1} I_K)$
2) For each of the C collapsed datasets (c.f. Section 4.3) that combines a target domain with a source domain,
   a) For each of the M documents,
      i) Choose a topic proportion θ_m from the distribution Dirichlet(α)
      ii) Choose the document offset ϵ_m from $\mathcal{N}(\mu_c, \Sigma_c)$
      iii) Set document latent vector as $v_m = \epsilon_m + \theta_m$
      iv) For each of the N activities,
         A) Choose topic id z_{mn} from Mult(θ_m)
         B) Choose an activity a_{mn} from Mult$(\beta_{z_{mn}})$
      v) For each existing user-scene annotation (i, m),
         A) Choose an annotation $r_{i,m}$ from $\sim \mathcal{N}(u_i^T v_m, c_{im}^{-1})$

Similar approaches that also model user annotations as a probabilistic matrix factorization over users and items do not consider the presence of multiple domains and therefore lose the advantage of transferring existing labels to a target domain to improve prediction [10], [11].

A. Learning Parameters

Let's assume the topics $\beta_{1:K}$ are fixed, the posterior distribution of CPM $p(u, v, \theta, z|\Sigma, \lambda_u, \alpha, \beta)$ can be factorized as follows,

$$\prod_{i=1}^{I} p(u_i|\lambda_u) \prod_{m=1}^{M} p(r_{i,m}|u_i, v_m) p(v_m|\Sigma, \mu, \theta_m) p(\theta_m|\alpha) \prod_{n=1}^{N} p(z_n|\theta_m) \prod_{n=1}^{N} p(a_{m,n}|z_{m,n}, \beta_k)$$

This expression is computationally intractable as the cardinality $VxMxN$ is too large. Our goal is to maximize

[3] A prior for ϵ is considered a purely subjective assessment in the way how a target domain is related to a source domains. More expressive metrics to relate content across domains could be used such as the part of the day (morning, afternoon, or evening) when the recording took place.

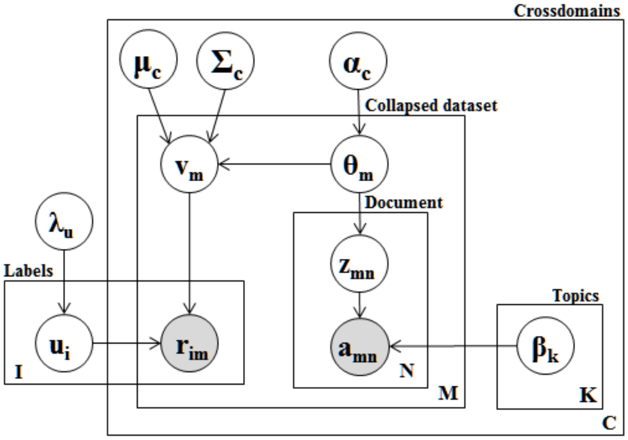


Fig. 6. Probabilistic Graphical Model of CPM. Activities, a_{mn}, and annotations, r_{im}, are observable variables shadowed in gray. Note that domain-independent topic proportions, θ_m, are defined for each collapsed dataset.

the posterior distribution by optimizing the functional dependence of the Gaussian on each hidden parameter. For example, the posterior of $p(v_m|\mu, \Sigma, \theta_m)$ has the quadratic form,

$$(v_m - (\theta_m + \mu_m))^T \Sigma_m (v_m - (\theta_m + \mu_m))$$

because each topic proportion is biased in response to a dissimilarity metric encoded in the covariance matrix Σ, $v_m \sim \mathcal{N}(\theta_m + \mu, \Sigma)$. Similarly, the quadratic form of the user annotation $p(r_{i,m}|u_i, v_m)$ is defined as,

$$(r_{i,m} - (u_i^T v_m))^T c_{i,m} (r_{i,m} - (u_i^T v_m))$$

The log likelihood of the posterior represents this value as the sum of probabilistic components,

$$\begin{aligned}
&\mathcal{L} = -\tfrac{\lambda_u}{2} \textstyle\sum_i u_i^T u_i \\
&-\tfrac{1}{2} \textstyle\sum_m (v_m - (\theta_m + \mu_m))^T \Sigma (v_m - (\theta_m + \mu_m)) \\
&- \textstyle\sum_{i,m} (r_{i,m} - \tfrac{1}{2}(u_i^T v_m))^T c_{i,m} (r_{i,m} - (u_i^T v_m)) \\
&+ \textstyle\sum_m \sum_n log(\theta_{jk} \beta_{k,w_{jn}})
\end{aligned}$$

First, we optimize $\mathcal{L}$ in terms of U and V and set them to zero to find their optimal parameters, similar to [12]. Then, we derive $\mathcal{L}$ in terms of θ to learn the topic proportions for each document and apply Jensen's inequality to provide a lower bound in terms of a function $q(\theta)$ that can be factorized with K independent components as $q(\theta) = \prod_i^K q(\theta_i)$. This result in the following factorization, $q(\theta) = q_1(\theta_{mk}) q_2(\beta_{k,w_{mn}})$, which can be optimized by coordinate ascend to find their optimum parameters by fixing one of them at each iteration.

VI. Experiments

In this section, we test the performance of CPM with surveillance videos recording activities in traffic roads. Moving objects describe traffic scenes governed by the state of multiple semaphores. We experiment on the following datasets: ***Street Intersection***[4] (*normal quality, 25fps, 90 minutes, 5 semaphores*), ***Karl-Wilhelm*** & ***Strabe Streets***[5] (*normal definition, 25fps, 2 hour, 3 semaphores*), and ***Roundabout Junction***[6] (*normal quality, 25fps, 2 hour, 3 semaphores*).

TABLE I. Information of the Training of Each Dataset.

Dataset	# Activities	Time to Compute
Street Intersection (5 semaphores)	318	4.21 hours
Karl-Wilhelm & Strabe (3 semaphores)	227	4.37 hours
Roundabout Junction (3 semaphores)	235	4.43 hours

We generate a ground truth in every dataset by asking users to manually tag scenes (time windows) of 10 seconds with either “dangerous” or “safe”. We generate domains in each dataset by considering three consecutive source domains and a target domain in the beginning and the end of each video, respectively. Every domain spans over 20 minutes. This generates 360 scenes for the source domain and 120 scenes for the target domain. To compute precision, we assume that no labels are available in the target domain and consider a hit if the average value of the precision with respect to the number of users is larger than 0.5 and a miss otherwise. For the generation of dictionaries of activities in each video, we use the same hash function in Timeseries Sensitive Hashing (TSH). The number of activities found in each video depends on the complexity of the scene, as shown in Table I. The more traffic lights yields more segmented activities and a larger dictionary. The experiments are run on a 3.6 GHz Pentium 4 with 4 GB RAM and all the above datasets are publicly available to facilitate later experimental comparisons.

A. Experiment 1: Finding Parameters

Before studying the performance of the algorithm, we analyze in this experiment the optimal values of external variables K (number of topics) and Σ (covariance between a source and target domain) for each dataset considering all the datasets.

First, we want to study the effect of changing the numbers of topics K in the model. We fix the other input parameters $\{\alpha = 0.1,\ \lambda_u = 0.5,\ \Sigma = 0.3, 0.3, 0.3\}$ and then compute the Mean Average Precision (MAP@20) for the scenes in the target domain with a increasing number of topics starting at 10. Figure 7 shows that a value of $K = 50$ topics provides a consistent result for all the datasets. Note that, the Street Intersection dataset shows the highest precision as it also contains the largest total number of activities. This is because scenes are better defined by topics that models co-occurrence with enough number of discrete activities.

[4] http://www.eecs.qmul.ac.uk/~jianli/Junction.html

[5] http://i21www.ira.uka.de/image_sequences/

[6] http://www.eecs.qmul.ac.uk/~jianli/Roundabout.html

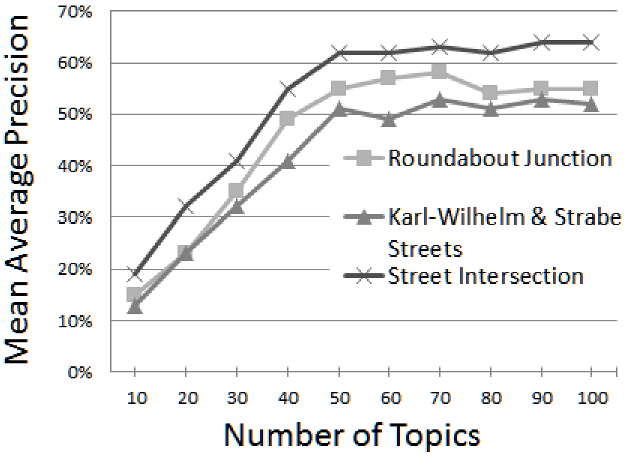


Fig. 7. Finding the optimal number of topics for each videoset in CPM.

Second, we observe the effect of the covariance matrix to relate different combinations of source and target domain in a video. We choose the Roundabout Junction dataset for this experiment because it is the largest, so it is more likely to find multiple domains over the video. We fix this time the number of topics to $K = 50$, in response to the above experiment, and the other input parameters as $\{\alpha = 0.1, \lambda_u = 0.5\}$. Then we compute the Mean Average Precision (MAP@20) for the scenes in the target domain, but considering each of the three source domains independently. Only one component of the covariance matrix is activated at a time to distinguish its entire effect (e.g., when considering the first source domain, only $\Sigma_{d=1}$ is activated). Figure 8 shows that the third source domain gets the highest precision with a value of $\Sigma_{d=3} = 0.8$. The choice of the last domain makes sense as it is the closest in time and thus the most likely to contain similar distributions of latent variables. Note that this effect may vary in much larger videos.

B. Experiment 2: Average Prediction in Target Domain

To make a competitive comparison, we evaluate the performance of CPM with the optimal K^* and Σ^* parameters computed in Experiment 1 and $\{\alpha = 0.1, \lambda_u = 0.5\}$, in terms of the Mean Average Precision (P@20) for predicting the true label in the target domain. This experiment examines the following methods:

a) SVM (trained with common feature-level variables, as explained in Section IV-A)

b) SVM (trained with common topic-level variables, as explained in Section IV-B)

c) SVM (trained with common crossdomain variables, as explained in Section IV-C)

d) Collaborative Filtering (collapsing both source and target domains in a single matrix)

e) CPM (with common topic-level variables) and

f) CPM Recommendation (our approach)

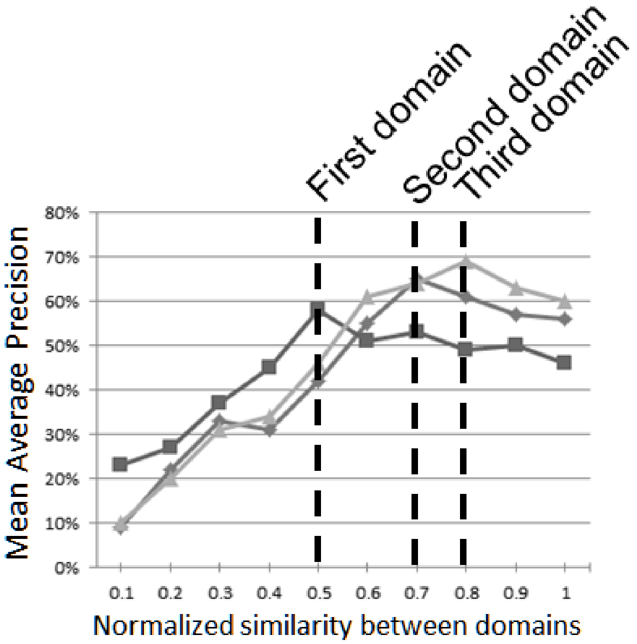


Fig. 8. Finding the optimal values in the covariance matrix that relates content with respect to each source domain.

As summarized in Figure 9, CPM shows better prediction of dangerous scenes in the target domain. The closest competitor is Collaborative Filtering, which does not consider any video content, but only user preferences on scenes. This shows the advantage of considering both content and user annotations for predicting activities.

On the other hand, SVM cannot make adequate generalization as the underlying distributions of random variables change for different domains (feature and topic-level variables). Moreover, when we train SVM with topics that model co-occurrences in two domains jointly, we observe the best performance of SVM for this problem. This confirms the initial hypothesis stated in the paper, the variable distribution of random variables that explains content in video negatively affects the generalization properties of supervised algorithms. This is more evident when the temporal gap between source and target domains is large as in the Karl-Wilhelm & Strabe Streets and Roundabout Junction datasets. However, when the temporal distance between both is small, the performance of SVM with crossdomain variables is similar to Collaborative Filtering and CPM with topic-level variables.

Finally, we experiment with a variation of CPM with topic-level variables, which transfer user annotations, but considers only topics that model activities in the source domain. The results in terms of MAP@20 are not as good as the CPM technique discussed in this paper and CF.

VII. Conclusion

The rise in the number of surveillance cameras has led to an increasing need to process surveillance video and understand the interactions of moving objects in real time. Because of the massive amount of data, fast, approximate

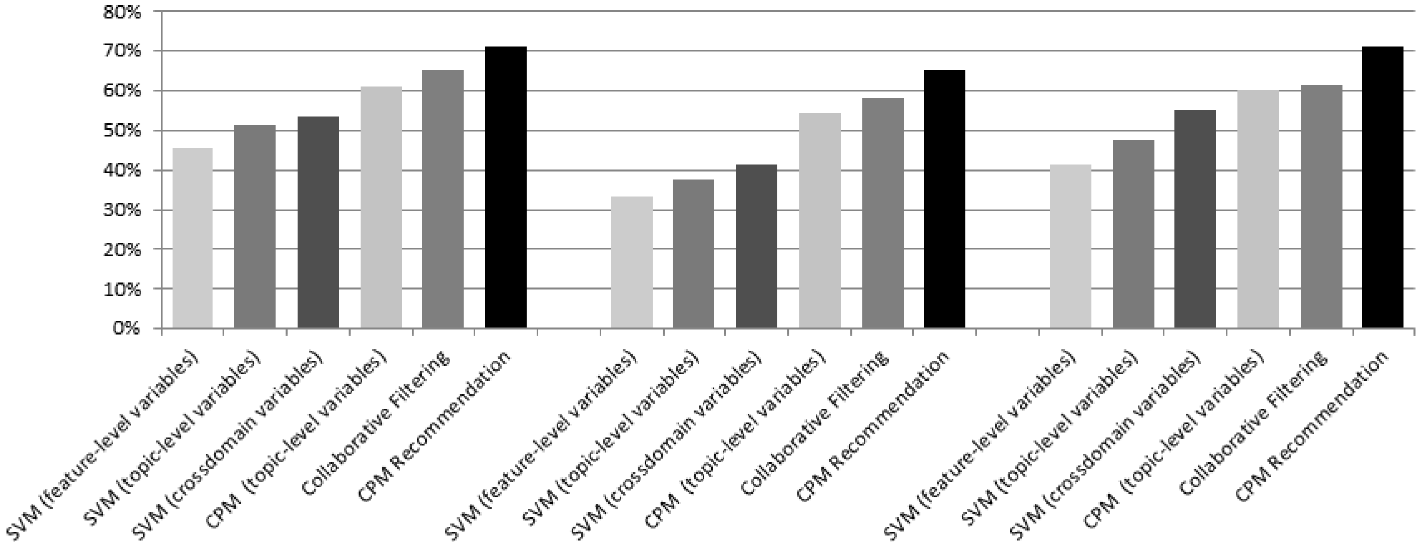


Fig. 9. Comparison of prediction techniques in terms of Mean Average Precision@20. a) SVM (trained with feature-level variables) (b) SVM (trained with topic-level variables) (c) SVM (trained with crossdomain variables) (d) Collaborative Filtering (e) CPM (with topic-level variables) and (f) CPM Recommendation

techniques based on statistical models are common. In this paper, we propose a new, cross-domain model that has uses labeled content from one segment of a video (source domains) to improve the prediction of user annotations in other segments (target domains). In our model, users annotate the source domains to describe events of interest in the domain. As the surveillance continues, new video with unknown content arrives to be processed. The model predicts the annotations in the newly arrived video based on the similarity of scenes between the source and target domains. Probabilistically, two scenes are similar if they share a set of random (observable or latent) variables. Experiments show that our method improves state-of-the-art techniques in predicting dangerous scenes in real-world traffic surveillance videos.

ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 1144404 entitled “III: EAGER: Aspect-oriented Data Weaving.” Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent dirichlet allocation,” *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, March 2003.

[2] O. A. B. Penatti, L. T. Li, J. Almeida, and R. da S. Torres, “A visual approach for video geocoding using bag-of-scenes,” in *Proceedings of the 2nd ACM International Conference on Multimedia Retrieval*, ser. ICMR ’12. New York, NY, USA: ACM, 2012, pp. 53:1–53:8.

[3] R. Yonetani, “Modeling video viewing behaviors for viewer state estimation,” in *Proceedings of the 20th ACM international conference on Multimedia*, ser. MM ’12. New York, NY, USA: ACM, 2012, pp. 1393–1396.

[4] O. U. Florez, A. Ocsa, and C. Dyreson, “Sublinear querying of realistic timeseries and its application to human motion,” in *Proceedings of the international conference on Multimedia information retrieval*. ACM, 2010, pp. 137–146.

[5] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, “Hierarchical Dirichlet Processes,” *Journal of the American Statistical Association*, vol. 101, pp. 1566–1581, 2006.

[6] T. Hospedales, S. Gong, and T. Xiang, “A markov clustering topic model for mining behaviour in video,” in *International Conference on Computer Vision*, 2009, pp. 1165–1172.

[7] X. Wang, X. Ma, and W. E. L. Grimson, “Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 3, pp. 539–555, 2009.

[8] J. Yin and Y. Meng, “Human activity recognition in video using a hierarchical probabilistic latent model,” in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2010, pp. 15–20.

[9] J. Li, S. Gong, and T. Xiang, “Global behaviour inference using probabilistic latent semantic analysis,” in *British Machine Vision Conference*, 2008, 5.

[10] H. Shan and A. Banerjee, “Generalized probabilistic matrix factorizations for collaborative filtering,” in *ICDM*, 2010, pp. 1025–1030.

[11] C. Wang and D. M. Blei, “Collaborative topic modeling for recommending scientific articles,” in *Proceedings of the 17th ACM SIGKDD conference*. New York, NY, USA: ACM, 2011, pp. 448–456.

[12] Y. Hu, Y. Koren, and C. Volinsky, “Collaborative filtering for implicit feedback datasets,” in *Proceedings of the 2008 ICDM*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 263–272.