LDA Topic Model with Soft Assignment of Descriptors to Words

Daphna Weinshall DAPHNA@CS.HUJI.AC.IL
Dmitri Hanukaev DMITRI.HANUKAEV@MAIL.HUJI.AC.IL
Gal Levi GAL.LEVI@MAIL.HUJI.AC.IL
School of Computer Science and Engineering and the Center for Neural Computation, The Hebrew University of Jerusalem, Jerusalem, Israel 91904

Abstract

The LDA topic model is being used to model corpora of documents that can be represented by bags of words. Here we extend the LDA model to deal with documents that are represented by bags of continuous descriptors. Given a finite dictionary of words, our extended LDA model allows for the soft assignment of descriptors to (many) dictionary words. We derive variational inference and parameter estimation procedures for the extended model, which closely resemble those obtained for the original model, with two important differences: First, the histogram of word counts is replaced by a histogram of pseudo word counts, or sums of responsibilities over all descriptors. Second, parameter estimation now depends on the average covariance matrix between these pseudo-counts, reflecting the fact that with soft assignment words are not independent. We use this approach to address the detection of novel video events, where we seek to identify video events with low posterior probability. Using a benchmark dataset for novelty detection, we show a very significant improvement in the detection of novel events when using our extended LDA model with soft assignment to words as against hard assignment (the original model), achieving state of the art novelty detection results.

1. Introduction

This work is motivated by the following general question: how do we catalogue a collection of multi media documents of continuous data (such as still images, video or music), discovering along the way the underlying structure in order to address such tasks as summarization, similarity computation, and novelty detection? We are particularly interested in novelty detection, where the task is to decide whether a new document, never seen before, is unlike any other document in the training sample. One way to address this question for discrete data employs the generative LDA model.

Latent Dirichlet Allocation (Blei et al., 2003) is a generative probabilistic model which recovers structure from a collection of discrete data, such as text documents. Documents are represented as bags of words, maintaining only the count of each word in the document (thus ignoring the order of words). The generative process of document sampling assumes a set of topics, where each document is sampled from a mixture of topics, and each topic defines some unique multinomial probability over the words in the dictionary. When fitting a corpus of documents with the LDA model, the topics which are discovered often reveal insightful information about the relations and shared structure between documents.

The LDA model has proven very useful for modeling text documents. With other documents such as images and video, it has been noted that these documents can often be described by bags of descriptors, where a descriptor is typically represented by a vector in $\mathcal{R}^d$ for some d. In order to get a 'bag of words' from such 'bag of descriptors', one typically starts by defining a dictionary of descriptors, obtained by vector quantization (or clustering) of the set of observed descriptors. Now each descriptor in the bag can be assigned a word (using, e.g., nearest neighbor in $\mathcal{R}^d$), and a bag of words representation is obtained (Csurka et al., 2004). This procedure is very effective, and has led to intriguing use of the basic LDA model and its extensions with non-textual data, see for example (Sivic et al., 2005;

Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28. Copyright 2013 by the author(s).

Fei-Fei & Perona, 2005; Sivic et al., 2008; Wang et al., 2009; Tuytelaars et al., 2010; Philbin et al., 2011) and (Wang & Grimson, 2007; Sudderth et al., 2005; Cao & Fei-Fei, 2007) for extensions which consider the spatial location of image patches.

However, the assignment of each descriptor to a single word can be problematic, and often it seems more appropriate to describe a descriptor by a mixture probability over words (Farquhar et al., 2005; Perronnin, 2008; Philbin et al., 2008; van Gemert et al., 2010). In particular, when one is interested in the modeling of new documents never seen before, as we do here, some descriptors in the bag of the new document may not correspond well to any word in the dictionary, but can still be effectively modeled by a mixture of words.

Can we still use LDA to model documents when this soft assignment representation is used? This is the technical question we address in this paper. In Section 2 we describe the extended LDA model, where documents are described as bags of continuous vector descriptors in $\mathcal{R}^d$, and each descriptor is modeled by a mixture probability over words. Thus defined, inference in this model can be achieved using MC methods and Gibbs sampling (Griffiths & Steyvers, 2004). Pursuing the complementary path and deterministic approximations, in Section 3 we derive effective variational algorithms for inference and parameter estimation in this model.

Our method is a generalization of the variational inference and parameter estimation algorithms described in (Blei et al., 2003) of similar low complexity. The main difference is that pseudo-counts of words (or summed responsibility) take the role of word counts (bin histograms), and the derivation requires further approximations as discussed in Section 3.2. Interestingly, for parameter estimation, our algorithm uses in addition the average covariance matrix between the pseudo-count vectors. Consequently inference and parameter estimation depend on both the first and second order statistics of pseudo-counts of words in the document.

In this work it is assumed that the dictionary is given - it is a collection of words, each defining a probability distribution over the space $\mathcal{R}^d$ of descriptors. This assumption may be questioned, in particular in light of the body of work on dictionary learning in the context of sparse coding which may be of relevance here (Olshausen et al., 1997; Ramirez et al., 2010). It may seem appealing to define a single framework where dictionary learning and topic modeling are done simultaneously, as is done for example in (Larlus & Jurie, 2009). (Rematas et al., 2012) goes a step further and models the distribution of descriptors non-parametrically. Both methods solve more difficult problems and rely on stochastic methods and Gibbs sampling, with many local minima and the ensuing dependence on initialization.

We chose to separate the problems and assume a pre-computed dictionary for a number of reasons. The primary reason is computational. On the one hand, simultaneous dictionary learning and topic modeling poses a rather difficult non convex optimization problem. On the other hand, given soft assignment to words in the dictionary, it would seem that any dictionary which can approximate the distribution of descriptors effectively via a mixture model will do well enough for the task. We therefore chose to use an approximate (possibly non optimal) dictionary and spend more effort on topic learning with soft assignment to words. The second reason lies in our application - novelty detection; optimal dictionary for the training data may not be optimal for truly novel examples.

In this work, therefore, dictionary learning is done during pre-processing of the training data. The dictionary is a set of words which act as the components of a mixture distribution, and the task of dictionary learning boils down to the task of estimating the components of a mixture distribution which models the empirical distribution of the observed descriptors in the training set. The distribution of each component (or word) can be estimated using, for example, a generative parametric approach such as Gaussian Mixture Modeling. In the application pursued in this paper we describe experiments with one such approach, and an alternative one which employs generative words and models descriptors with a generative probabilistic model called dynamic texture (Doretto et al., 2003). In the latter approach dictionary learning resorts to the selection of a fixed number of DT's, which can generate with high likelihood most of the observed video events (Chan & Vasconcelos, 2008).

It is not easy to evaluate the efficacy and success of topic models, since there is no ground truth of actual topics to compare with. We chose to evaluate our method in the task of novelty detection. In Section 4 we use the method described above to model video data and identify novel video events. Video events are represented using a dictionary of ISA or dynamic texture features. We test our ability to identify novel events using a benchmark dataset designed for the evaluation of novelty detection algorithms (Mahadevan et al., 2012), demonstrating significant improvement obtained by soft assignment as compared with hard assignment (for the same dictionary).

2. Extended LDA Model

2.1. Notations

We use notations very similar to the language of text collections, making the necessary distinctions along the way. Our goal is to provide a generative model for a collection of documents, each represented by a set of un-ordered descriptors. Like in the bag of words model, the bag of descriptors model ignores the order and relative location of descriptors. Unlike the bag of words model, a descriptor is not a discrete word but a continuous measurement, whose distribution we model by a generative process as well - a mixture over a set of discrete components which we will henceforce call words. In other words, the given dictionary is a collection of generative word models (aka components), and each word assigns a probability to every measured descriptor (aka responsibility).

Formally, the model is described as follows:

- A *descriptor* x is a basic unit of continuous data. It is modeled by a vector of probabilities, based on V categorical distributions $\{w_j,\ 1 \leq j \leq V\}$ that we call *words*. Thus we represent a descriptor by a vector $\mathbf{f} \in \mathcal{R}^V$, where element $f_j,\ 0 \leq f_j \leq 1$ denotes the probability that the descriptor has been generated by word w_j. This probability function is known apriori, and it is part of the given dictionary.
- A *document* is a sequence of N descriptors $\mathbf{x} = \{x_1, x_2, \ldots, x_N\}$, represented by their probability vectors $\mathcal{F} = \{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_N\}$.
- A *corpus* is a collection of M documents represented by $\mathcal{D} = \{\mathcal{F}_1, \mathcal{F}_2, \ldots \mathcal{F}_M\}$.

As in other topic models, we aim to find a probabilistic model of the corpus that assigns high probability to similar documents which we have not yet seen.

2.2. LDA with Soft Word Assignment

In many ways, our model is similar to the original LDA model described in (Blei et al., 2003). Documents are represented as mixtures over latent 'topics', and each topic is characterized by a distribution over 'words'. There is one additional step, where the descriptor is sampled from the generative word model. Thus the model assumes the following generative process for each document $\mathcal{F}$ in the corpus (see Fig. 1):

1. Choose probability coefficients over topics $\theta \sim Dir(\alpha)$.
2. For each of the N descriptors x_n in the bag of descriptors for document $\mathcal{F}$:
   (a) Choose one of k topics $z_n \sim Multinomial(\theta)$.
   (b) Choose one of V words w_j from the multinomial distribution $p(w \mid z_n, \beta)$.
   (c) Choose descriptor x_n from the conditional distribution $p(x_n \mid w_j)$ defined by the generative word model of w_j.

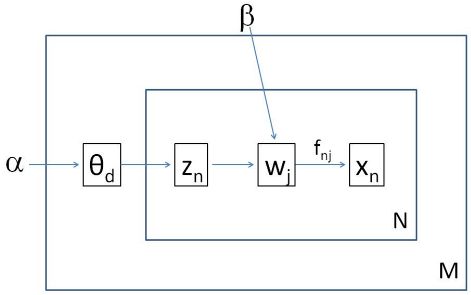

Figure 1. The generative Latent Dirichlet Allocation model with generative words.

The difference between this model and the original LDA model lies in step 2c, the last box in Fig. 1. Unlike before, each descriptor is not assigned to a single word, but is itself generated by a mixture model of discrete words. Thus, the probability of the observation x_n is the mixture probability $p(x_n \mid z_n, \beta) = \sum_{j=1}^{V} p(w_j \mid z_n = i, \beta) p(x_n \mid w_j) = \sum_j \beta_{ij} p(x_n \mid w_j)$.

Importantly, we assume that the dictionary is given, in terms of the mixture components. The dictionary is therefore a list of V probability functions, from which we compute $f_{nj} = p(x_n \mid w_j)$; here x_n are the observables, $\alpha,\ \beta$ are the model's parameters, and $\theta,\ \mathbf{z},\ \mathbf{w}$ its hidden variables.

Given the parameters $\alpha,\ \beta$, the joint distribution of the observables and the hidden variables is:

$$p(\theta, \mathbf{z}, \mathbf{w}, \mathbf{x} \mid \alpha, \beta) = p(\theta \mid \alpha) \prod_{n=1}^{N} p(z_n \mid \theta) p(w_j \mid z_n, \beta) p(x_n \mid w_j) \qquad (1)$$

The probability of the corpus is the following

$$p(\mathcal{D} \mid \alpha,\beta) = \prod_{d=1}^{M} \int p(\theta_d \mid \alpha) \left(\prod_{n=1}^{N_d} \sum_{z_{dn}} \sum_{j} p(z_{dn} \mid \theta_d) p(w_j \mid z_{dn}, \beta) p(x_{dn} \mid w_j) \right) d\theta_d$$

3. Inference and Parameter Estimation

We now describe variational inference and parameter estimation algorithms for the model described above, extending the procedures described in (Blei et al., 2003) and using the same variational parameters. Interestingly, the ensuing inference algorithm gives misleadingly similar update rules of similar complexity. For parameter estimation we provide a new lower bound which, when maximized, once again gives update rules very similar to those obtained for the original LDA model. Finally, we show how inference and parameter estimation can be efficiently used with a representation similar to bag of words, where each word is represented by its pseudo-count - the sum of its likelihood over all descriptors in the document.

We start by stating our goal, which is to estimate the probability

$$p(\mathbf{x} \mid \alpha, \beta) = \int \sum_{\mathbf{z}} \sum_{\mathbf{w}} p(\theta, \mathbf{z}, \mathbf{w}, \mathbf{x} \mid \alpha, \beta) d\theta$$

Estimating this probability is intractable, and we follow (Blei et al., 2003) and their choice of variational inference to approximate this function.

3.1. Variational Inference

Since, given descriptor x_n, the dictionary provides the vector of conditional probabilities $\mathbf{f}_n$ where $f_{nj} = p(x_n \mid w_j)$, we marginalize over the hidden variable $\mathbf{w}$ and attempt to infer θ, $\mathbf{z}$. To accomplish this goal, we define the following variational distribution

$$q(\theta, \mathbf{z} \mid \gamma, \phi) = q(\theta \mid \gamma) \prod_{n=1}^{N} q(z_n \mid \phi_n) \tag{2}$$

where $q(\theta) \sim Dir(\gamma)$ and $q(z_n) \sim Multinomial(\phi_n)$. The function $q(\theta, \mathbf{z} \mid \gamma, \phi)$ is a surrogate for the posterior distribution (marginalized over $\mathbf{w}$) $p(\theta, \mathbf{z}, \mathbf{x} \mid \alpha, \beta)$.

Following (Blei et al., 2003)), we use Jensen inequality to obtain a lower bound on $\log p(\mathbf{x} \mid \alpha, \beta)$ for any surrogate function $q(\theta, \mathbf{z} \mid \gamma, \phi)$:

$$\begin{aligned}\log\, p(\mathbf{x} \mid \alpha, \beta) &= \log \int \sum_{\mathbf{z}} \frac{p(\theta, \mathbf{z}, \mathbf{x} \mid \alpha, \beta)}{q(\theta, \mathbf{z})} q(\theta, \mathbf{z}) d\theta \\ &= \log E_q \left[\frac{p(\theta, \mathbf{z}, \mathbf{x} \mid \alpha, \beta)}{q(\theta, \mathbf{z})} \right] \\ &\geq E_q[\log p(\theta, \mathbf{z}, \mathbf{x} \mid \alpha, \beta)] - E_q[\log q(\theta, \mathbf{z})] \\ &= \mathcal{L}(\gamma, \phi; \alpha, \beta)\end{aligned}$$

In order to approximate $\log p(\mathbf{x} \mid \alpha, \beta)$, one maximizes the lower bound $\mathcal{L}(\gamma, \phi; \alpha, \beta)$ with respect to all parameters $\gamma, \phi, \alpha, \beta$. Specifically, let

$$\begin{aligned}\mathcal{L}(\gamma, \phi; \alpha, \beta) = & E_q[\log p(\theta \mid \alpha)] + E_q[\log p(\mathbf{z} \mid \theta)] \\ & + E_q[\log p(\mathbf{x} \mid z, \beta)] \\ & - E_q[\log q(\theta)] - E_q[\log q(\mathbf{z})]\end{aligned} \tag{3}$$

This expression is almost identical to the one we get with the original LDA model, with one difference in the third term above:

$$E_q[\log p(\mathbf{x} \mid z, \beta)] = \sum_{n=1}^{N} \sum_{i=1}^{k} \phi_{ni} \log[\sum_{j=1}^{V} \beta_{ij} f_{nj}] \tag{4}$$

After substituting the expressions for the functions p, q, we may take derivatives with respect to the variational parameters and set them to 0, thus obtaining:

$$\phi_{ni} \propto (\sum_{j=1}^{V} \beta_{ij} f_{nj}) \exp(\Psi(\gamma_i) - \Psi(\sum_{l=1}^{k} \gamma_l)) \tag{5}$$

$$\gamma_i = \alpha_i + \sum_{n=1}^{N} \phi_{ni} \tag{6}$$

where Ψ is the digamma function.

3.2. Parameter Estimation

Next, we wish to estimate the model parameters α, β given corpus $\mathcal{D}$ following the same Bayesian procedure and variational approximation. Differentiating (3) with respect to α, one still gets

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \alpha_i} =& M \left(\Psi(\sum_{l=1}^{k} \alpha_l) - \Psi(\alpha_i) \right) + \\ & \sum_{d=1}^{M} \left(\Psi(\gamma_{di}) - \Psi(\sum_{l=1}^{k} \gamma_{dl}) \right)\end{aligned}$$

which can be solved with the Newton-Raphson algorithm as in (Blei et al., 2003).

In order to compute β, we need to maximize (4) summed over all the documents, and subject to the constraint that the columns of β sum to 1. Adding Lagrange multipliers, a solution can be obtained by differentiating

$$\sum_{d=1}^{M} \sum_{n=1}^{N_d} \sum_{i=1}^{k} \phi_{dni} \log[\sum_{j=1}^{V} \beta_{ij} f_{dnj}] + \sum_{i=1}^{k} \mu_i (\sum_{j=1}^{V} \beta_{ij} - 1) \tag{7}$$

with respect to β_{ij} and μ_i, and setting the derivatives to 0. This turns out to be hard to do in closed form, and the solution can be approximated using gradient descent.

Instead, we derive a lower bound on the function we aim to maximize. Using the concavity of the log function, it follows that

$$\log[\sum_{j=1}^{V}\beta_{ij}f_{dnj}] = \log[(\sum_{a=1}^{V}f_{dna})(\sum_{j=1}^{V}\beta_{ij}\frac{f_{dnj}}{\sum_{a=1}^{V}f_{dna}})]$$

$$\geq \log(F_{dn}) + \sum_{j=1}^{V}\frac{f_{dnj}}{F_{dn}}\log\beta_{ij} \qquad (8)$$

where $F_{dn} = \sum_{a=1}^{V} f_{dna}$. We maximize this function with respect to β under the normalization constraint, to obtain

$$\beta_{ij} \propto \sum_{d=1}^{M}\sum_{n=1}^{N_d}\phi_{dni}\frac{f_{dnj}}{F_{dn}} \qquad (9)$$

It is now possible to improve this estimate of β_{ij} using gradient ascent with respect to (7).

3.3. Pseudo-count Histograms for Bags of Descriptors

When computing variational inference for the LDA model as described in (Blei et al., 2003), it follows from the exchangeability assumption that it is not necessary to keep an account of the actual list of words in the document, but only the count of words (or word histogram). In this section we will show that the same holds for our extended LDA model, where the count is replaced by 'pseudo-count' - the sum of word probabilities over the document.

In the original LDA model, we first note that due to exchangeability, ϕ_n for a given document (we omit the index d for clarity) is the same for all word locations n where the same word w_a is observed in the original bag of words model. We can therefore define a vector of length V $\hat{\phi}_a$, where $\phi_n = \hat{\phi}_a$ for every word 'a' such that $w_a^n = 1$ (i.e., word w_a is observed in location n). The update rule for $\hat{\phi}$ is:

$$\hat{\phi}_{ai} \propto \beta_{ia}\exp(\Psi(\gamma_i) - \Psi(\sum_{l=1}^{k}\gamma_l))$$

$$\Rightarrow \quad \hat{\phi}_{ai} = \frac{\beta_{ia}}{Q_a}\exp(\Psi(\gamma_i) - \Psi(\sum_{l=1}^{k}\gamma_l)) \qquad (10)$$

where $Q_a = \sum_{i=1}^{k}\beta_{ia}\exp(\Psi(\gamma_i) - \Psi(\sum_{l=1}^{k}\gamma_l))$. We can express both variational parameters in terms of $\hat{\phi}$, since $\gamma_i = \alpha_i + \sum_{a=1}^{V}\hat{\phi}_{ai}cnt(a)$ for $cnt(a)$ the number of times word 'a' appeared in the document. Thus, the original LDA variational inference algorithm essentially relies only on word counts in the document.

In the extended model, we can use $\hat{\phi}$ as defined in (10). A crude approximation allows us to derive very similar update rules for the variational parameters:

$$\phi_{ni} \approx \sum_{j=1}^{V}\hat{\phi}_{ji}\frac{f_{nj}}{F_n} \qquad (11)$$

where $F_n = \sum_{a=1}^{V} f_{na}$. This normalized vector approximates the expression in (5), and is true when β_{ij} is the same for all topics j. Now

$$\gamma_i = \alpha_i + \sum_{n=1}^{N}\phi_{ni} = \alpha_i + \sum_{n=1}^{N}\sum_{j=1}^{V}\hat{\phi}_{ji}\frac{f_{nj}}{F_n}$$

$$= \alpha_i + \sum_{a=1}^{V}\hat{\phi}_{ai}PseudoCnt(a)$$

where $PseudoCnt(a) = \sum_{n=1}^{N}\frac{f_{na}}{F_n}$.

As to be expected by now, parameter estimation is more complicated in the extended model. The computation of α depends only on the variational parameters, and can therefore be done using the histogram of pseudo-counts. The approximation of β in Eq. (9) can be done as follows:

$$\beta_{ij} \propto \sum_{d=1}^{M}\sum_{n=1}^{N_d}\phi_{dni}\frac{f_{dnj}}{F_{dn}}$$

$$= \sum_{d=1}^{M}\sum_{n=1}^{N_d}\sum_{a=1}^{V}\hat{\phi}_{dai}\frac{f_{dna}}{F_{dn}}\frac{f_{dnj}}{F_{dn}}$$

$$= \sum_{d=1}^{M}\sum_{a=1}^{V}\hat{\phi}_{dai}\sum_{n=1}^{N_d}\frac{f_{dna}}{F_{dn}}\frac{f_{dnj}}{F_{dn}} \qquad (12)$$

where (11) is used in the transition from the first to the second line.

Let A^d denote the average covariance matrix for the probability vectors $\frac{\mathbf{f}_{dn}}{F_{dn}}$:

$$A^d = \sum_{n=1}^{N}\frac{\mathbf{f}_{dn}}{F_{dn}} \cdot \frac{\mathbf{f}_{dn}^T}{F_{dn}}$$

Then (12) can be written as

$$\beta_{ij} \propto \sum_{d=1}^{M}\sum_{a=1}^{V}\hat{\phi}_{dai}A_{aj}^d$$

In vector notation, where $\vec{\beta}_i$ denotes the probabilities for word assignment in topic i and $\vec{\phi}_{di}$ the variational vector of probabilities for word assignment in document d for topic i,

$$\vec{\beta}_i \propto \sum_{d=1}^{M}A^d\vec{\phi}_{di} \qquad (13)$$

In conclusion, in order to infer the hidden variables of the modified LDA model with the variational approximation, we do not need to keep the vector $\mathbf{f}_n$ for each word, but only the histogram of word pseudo-counts - a histogram h where $h(a) = \sum_{n=1}^{N} \frac{f_{na}}{F_n}$ for $1 \leq a \leq V$ the index of the word in the dictionary. In order to approximate the model parameter β, we also need matrix A^d which measures the covariance between the words in document d.

3.4. Smoothing and LDA

In (Blei et al., 2003) Blei et al. described a slightly different model where they proposed to place a Dirichlet prior on the hyper parameter β. This model turned out to be better suited for online implementation (Hoffman et al., 2010), and we use it here for the same purpose with an online algorithm and the extended LDA model. In this model the probability of the data becomes:

$$p(\mathcal{D} \mid \alpha, \eta) = \int \prod_{i=1}^{k} p(\beta_i \mid \eta) p(\mathcal{D} \mid \alpha, \beta) d\beta \qquad (14)$$

where $p(\mathcal{D} \mid \alpha, \beta)$ corresponds to the LDA model described above.

Now, the variational distribution takes the form:

$$\begin{aligned} &q(\beta_{1:k}, \mathbf{z}_{1:M}, \theta_{1:M} | \lambda, \phi, \gamma) \\ &= \prod_{i=1}^{k} Dir(\beta_i | \lambda_i) \prod_{d=1}^{M} q_d(\theta_d, \mathbf{z}_d | \phi_d, \gamma_d), \end{aligned} \qquad (15)$$

It can easily be verified that the update equation for λ and ϕ become:

$$\lambda_{ij} = \eta + \sum_{d=1}^{M} A^d \vec{\phi}_{di},$$

$$\begin{aligned} \hat{\phi}_{ai} &\propto \exp(\log(\beta_{ia} | \lambda_i) \exp(\Psi(\gamma_i) - \Psi(\sum_{l=1}^{k} \gamma_l)) \qquad (16) \\ &= \exp(\Psi(\lambda_{ia}) - \Psi(\sum_{j=1}^{V} \lambda_j)) \exp(\Psi(\gamma_i) - \Psi(\sum_{l=1}^{k} \gamma_l)) \end{aligned}$$

4. Novelty Detection in Video Data

Successfully identifying novel video events can be very useful for such applications as video surveillance and video annotation. An effective way to identify novel events is to use extended recordings to learn the 'normal' state of the system, and then identify those events whose posterior probability with respect to the learnt model is low. In this section we use the extended LDA model described above to model the given collection of video recordings.

4.1. Representation of Video Events

In order to represent video events, each video document is decomposed into a set of spatio-temporal patches extracted from the video. Each spatio-temporal patch is represented via the ISA features (Le et al., 2011), or the generative *Dynamic Textures* model which readily assigns a probability to each *3D-patch*. The dictionary is a set of words pre selected based on the respective representation of all patches in the training data, and each spatio-temporal video patch is represented as a mixture of these categorical words. In estimating the model parameters, and in particular the parameter β, we effectively learn the mixture coefficients of the words in each topic.

More specifically, for ISA spatio-temporal features we use first layer features trained on Hollywood2 dataset as provided by the authors (Le et al., 2011). We obtain a dictionary with the k-means algorithm (Euclidean distance) using subset of the train set. We estimate $p(x|w)$ for each of the words in the dictionary by fitting a $1D$ Gaussian, using distances between features assigned to w by the k-means algorithm.

Dynamic Texture (DT) (Doretto et al., 2003) is a video generative model that captures both the dynamic and the texture of the video. It consists of a random process containing an *observed variable* y_t, which encodes the appearance component (video frame at time t), and a *hidden state* variable x_t, which encodes the dynamics (evolution of the video over time). The state and observed variables are related through the linear dynamical system defined by:

$$\begin{cases} x_{t+1} = Ax_t + v_t \\ y_t = Cx_t + w_t \end{cases} \qquad (17)$$

where $x_t \in \mathcal{R}^n$ denotes the *hidden state* variable, $y_t \in \mathcal{R}^m$ the observable, $A \in \mathcal{R}^{n \times n}$ the state transition matrix, and $C \in \mathcal{R}^{m \times n}$ an observation matrix. $v_t \in \mathcal{R}^n$ and $w_t \in \mathcal{R}^m$ denote additive zero-mean Gaussian noise in the hidden and the observed states.

Given a training set of video sequences, each sequence is divided into a set of video events which are object-size spatio-temporal video pieces, see Fig. 2. Each video events is further divided into small spatio-temporal *3D-patches*; these patches together define the ensemble of basic video units, which are used to learn the words in the dictionary using the *k-means* algorithm. With the DT representation we use the following initialization procedure: we pick V random *3D-patches*, learn for each patch the DT that maximizes its likelihood, and use these V DT's as initial centers for the *k-means* algorithm.

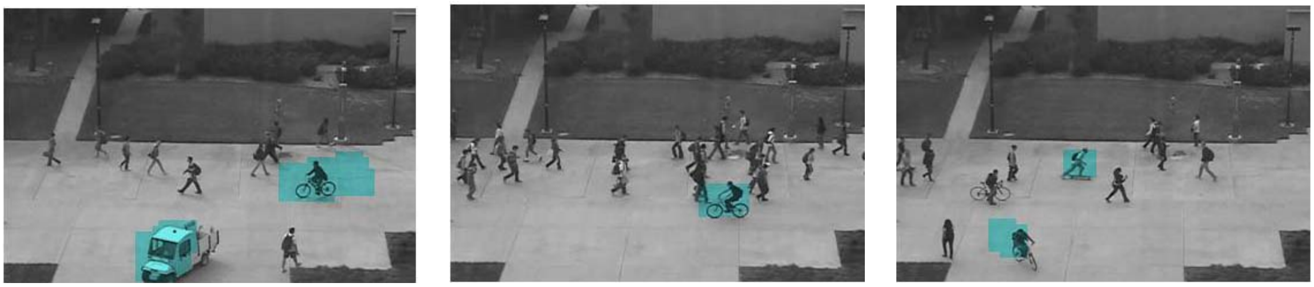

Figure 3. Examples of abnormal events.

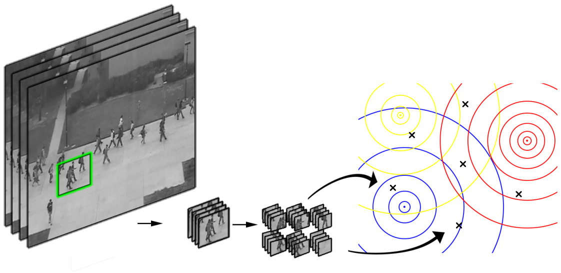

Figure 2. A video sequences (left) is divided into video events (marked with green border). Each event is divided to small spatio-temporal patches, and each patch is represented by a mixture of basic words.

4.2. Database

To evaluate our algorithm we used the UCSD Ped2 dataset (Mahadevan et al., 2012), consisting of videos taken while monitoring a crowded pedestrian walkway scene. The dataset consists of 16 training videos containing only pedestrians, and 12 test videos containing also abnormal events such as bicycles and skateboard riders; examples of abnormal events detected by our algorithm are shown in Fig. 3. The dataset also contains frame level ground truth of frames that contain abnormal events, and there is a subset of 9 videos provided with pixel-level binary masks.

Each video has 120-180 frames of resolution 360×240. We split each video to events of size $24 \times 24 \times 21$ (21 is the temporal length). With *DT* features each event is decomposed into $4 \times 4 \times 3$ *3D-patches* of size $9 \times 9 \times 10$, with a spatial/temporal displacement of 5 pixels/frames between each two events and *3D-patches*. With ISA features each event is decomposed into $3 \times 3 \times 4$ *3D-patches* of size $16 \times 16 \times 10$ (Le et al., 2011). To improve performance we filtered out events and *3D-patches* with no movement. From the collection *3D-patches* a dictionary was computed.

When determining the dictionary size, we seek the smallest dictionary which will allow for good performance and will benefit generalization. At the same time, hard assignment to words would appear to benefit from a larger dictionary size when competing with the richer soft assignment to words. We therefore choose the size of the dictionary based on the performance of the hard assignment algorithm on the train set, searching for the best size over a feasible range. For both feature types the optimal dictionary size was 75 words. With *DT* words we used *hidden state* size of 10 learnt as described in Section 4.1, see Fig. 4.

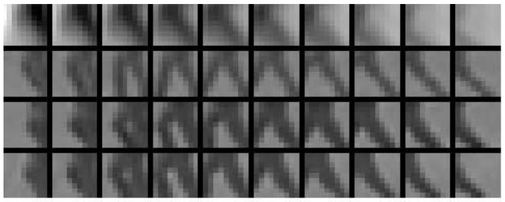

Figure 4. Illustration of one *DT* word, which captures a leg moving to the left (frames are ordered from left to right). First row: a synthetic *3D-patch* generated from the *DT*. Subsequent rows: three *3D-patches* whose most likely word assignment was identified as this *DT*.

4.3. LDA Model and Results

Given the dictionary, each *3D-patch* was assigned a vector of soft probabilities, reflecting the likelihood that each word generated the patch. In soft assignment of this nature it is customary to set to zero some of the lower assignment, typically those below the average assignment (Coates et al., 2010). Here it was necessary to use a higher threshold and set to zero up to 95% of the lower assignments, because the *DT* dictionary words in particular have high overlap. Next, each video event in the dataset was encoded according to the *3D-patches* composing it as described in Section 2. Finally, using events from the training video an extended LDA model was learnt as described in Section 3, and subsequently used to estimate the likelihood of each event in the test videos.

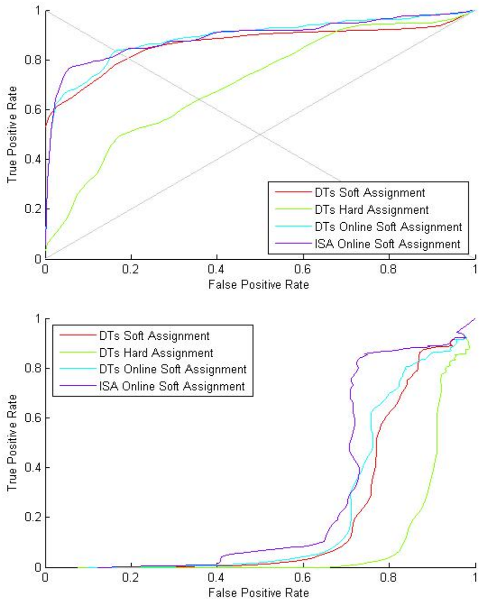

Figure 5. ROC curves for frame level analysis (top) and pixel level (bottom).

Using event likelihood, we identify those events whose likelihood is below some threshold as abnormal events. Varying the threshold value, we obtained the ROC curves shown in Fig. 5. Following (Mahadevan et al., 2010), we tested our method using frame-level and pixel-level measurements, where in the former a detection in a frame is successful regardless of the abnormality location within the frame, and in the later a detection is successful if at least 40% of the truly anomalous pixels are detected. The frame level EER values, compared with the results reported in (Mahadevan et al., 2010), are tabulated in Table 1. Some examples of successful detection are displayed in Fig. 3.

method	EER
SF (Mahadevan et al., 2010)	42%
MPPCA (Mahadevan et al., 2010)	30%
SF-MPPCA (Mahadevan et al., 2010)	36%
Adam (Adam et al., 2008)	42%
MDT (Mahadevan et al., 2010)	25%
Hard assignment, DT batch	35%
Soft assignment, DT batch	20%
Soft assignment, DT online	16%
Soft assignment, ISA online	17.6%

Table 1. Frame level EER values for different methods, see text.

We report results with hard assignment to words, soft assignment to words with the batch LDA algorithm described in Sections 3.1-3.2, and soft assignment to word with an online LDA algorithm (Hoffman et al., 2010) which computes the smooth LDA model with pseudo-count representation as outlined in Section 3.4. The online algorithm used both types of features, DT and ISA. In our comparisons soft assignment achieved top performance, better than alternative methods and much better than hard assignment.

4.4. Discussion

The most interesting result for evaluating the added value of the new model presented in this paper, is seen in the big difference in performance between soft and hard assignments of visual words. Using the same dictionary and all else being equal, hard assignment of video patches to *words* and the original LDA model achieved poor performance in novelty detection, while soft assignment and the extended LDA model (either batch or online) achieved state of the art performance. Moreover, this result was seen for two very different kinds of representation, one with generative feature model (DT) and one with an adaptive model trained on a different dataset (ISA). We note that the number of soft assignments had to be restricted in order for the batch LDA algorithm to converge to a non-trivial solution (i.e., non degenerate topic distribution).

5. Summary

Latent Dirichlet Allocation has proven very effective in modeling text and other multi-media documents. Here we extended this model to allow for a richer representation, to answer a need when dealing with videos and images which are less naturally represented by bags of words. Documents are now represented by bags of continuous descriptors, and each descriptor is represented by its vector of affinities to the words in the dictionary. We derived variational inference and parameter estimation procedures for this model, which resemble the original algorithm in an appealing way. We demonstrated how the incorporation of soft assignment to words very significantly improved the effectiveness of the LDA model in the detection of novel video events. Specifically, when using the same data and the same dictionary, the traditional LDA model achieved poor performance in novelty detection, while our extended LDA method achieved state of the art performance.

Acknowledgements

This work was supported in part by grants from the Israel Science Foundation and the Intel Collaborative Research

Institute for Computational Intelligence (ICRI-CI).

References

Adam, A., Rivlin, E., Shimshoni, I., and Reinitz, D. Robust real-time unusual event detection using multiple fixed-location monitors. *IEEE Trans. PAMI*, 30(3):555–560, 2008.

Blei, D. M., Ng, A.Y., and Jordan, M.I. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.

Cao, L. and Fei-Fei, L. Spatially coherent latent topic model for concurrent segmentation and classification of objects and scenes. In *Proc. ICCV*, pp. 1–8, 2007.

Chan, A.B. and Vasconcelos, N. Modeling, clustering, and segmenting video with mixtures of dynamic textures. *IEEE Trans. PAMI*, 30(5):909–926, 2008.

Coates, A., Lee, H., and Ng, A.Y. An analysis of single-layer networks in unsupervised feature learning. *Ann Arbor*, 1001:48109, 2010.

Csurka, G., Dance, C., Fan, L., Willamowski, J., and Bray, C. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, pp. 22, 2004.

Doretto, G., Chiuso, A., Wu, Y.N., and Soatto, S. Dynamic texturbes. *International Journal of Computer Vision*, 51 (2):91–109, 2003.

Farquhar, J., Szedmak, S., Meng, H., and Shawe-Taylor, J. Improving "bag-of-keypoints" image categorisation: Generative models and pdf-kernels. 2005.

Fei-Fei, L. and Perona, P. A bayesian hierarchical model for learning natural scene categories. In *Proc. CVPR*, volume 2, pp. 524–531, 2005.

Griffiths, Thomas L and Steyvers, Mark. Finding scientific topics. *PNAS*, 101(Suppl 1):5228–5235, 2004.

Hoffman, Matthew, Blei, David M, and Bach, Francis. Online learning for latent dirichlet allocation. *Advances in Neural Information Processing Systems*, 23:856–864, 2010.

Larlus, D. and Jurie, F. Latent mixture vocabularies for object categorization and segmentation. *Image and Vision Computing*, 27(5):523–534, 2009.

Le, Quoc V, Zou, Will Y, Yeung, Serena Y, and Ng, Andrew Y. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *Proc. CVPR*, pp. 3361–3368, 2011.

Mahadevan, Vijay, Li, Weixin, Bhalodia, Viral, and Vasconcelos, Nuno. Anomaly detection in crowded scenes. In *Proc. CVPR*, 2010.

Mahadevan, Vijay, Li, Weixin, Bhalodia, Viral, and Vasconcelos, Nuno. http://www.svcl.ucsd.edu/projects/anomaly/dataset.html Accessed October 2011, 2012.

Olshausen, B.A., Field, D.J., et al. Sparse coding with an overcomplete basis set: A strategy employed by vi? *Vision research*, 37(23):3311–3326, 1997.

Perronnin, F. Universal and adapted vocabularies for generic visual categorization. *IEEE Trans. PAMI*, 30 (7):1243–1256, 2008.

Philbin, J., Chum, O., Isard, M., Sivic, J., and Zisserman, A. Lost in quantization: Improving particular object retrieval in large scale image databases. In *Proc. CVPR*, pp. 1–8, 2008.

Philbin, J., Sivic, J., and Zisserman, A. Geometric latent dirichlet allocation on a matching graph for large-scale image datasets. *International journal of computer vision*, 95(2):138–153, 2011.

Ramirez, I., Sprechmann, P., and Sapiro, G. Classification and clustering via dictionary learning with structured incoherence and shared features. In *Proc. CVPR*, pp. 3501–3508, 2010.

Rematas, K., Fritz, M., and Tuytelaars, T. Kernel density topic models: Visual topics without visual words. In *NIPS workshops, Modern Nonparametric Methods in Machine Learning*, 2012.

Sivic, J., Russell, B.C., Efros, A.A., Zisserman, A., and Freeman, W.T. Discovering objects and their location in images. In *Proc. ICCV*, volume 1, pp. 370–377, 2005.

Sivic, J., Russell, B.C., Zisserman, A., Freeman, W.T., and Efros, A.A. Unsupervised discovery of visual object class hierarchies. In *Proc. CVPR*, pp. 1–8, 2008.

Sudderth, E.B., Torralba, A., Freeman, W.T., and Willsky, A.S. Learning hierarchical models of scenes, objects, and parts. In *Proc. ICCV*, volume 2, pp. 1331–1338, 2005.

Tuytelaars, T., Lampert, C.H., Blaschko, M.B., and Buntine, W. Unsupervised object discovery: A comparison. *International Journal of Computer Vision*, 88(2):284–302, 2010.

van Gemert, J.C., Veenman, C.J., Smeulders, A.W.M., and Geusebroek, J.M. Visual word ambiguity. *IEEE Trans. PAMI*, 32(7):1271–1283, 2010.

Wang, C., Blei, D., and Li, F.F. Simultaneous image classification and annotation. In *Proc. CVPR*, pp. 1903–1910, 2009.

Wang, X. and Grimson, E. Spatial latent dirichlet allocation. *Advances in Neural Information Processing Systems*, 20:1577–1584, 2007.