Latent Dirichlet Allocation with Soft Assignment of Descriptors to Words

by

Gal Levl

under the supervision of
Prof. Daphna Weinshall

a thesis submitted in partial fulfillment of the
requirements for the degree of
Master of Science

at the School of Computer Science and Engineering
Hebrew University of Jerusalem, Israel 91904

June, 2013

Contents

1 Introduction **1**

1.1 Motivation 1

1.2 Our Approach 2

1.3 Related Work 2

2 Event Representation Using Bag Of Words **4**

2.1 Tracking-Free Pedestrian Events 4

2.2 Bag Of (Hard Assigned) Words 5

2.3 Bag Of Soft Assigned Words 7

3 Modeling of Typical Events Using LDA **9**

3.1 Latent Dirichlet Allocation 9

3.2 Inference 10

3.3 Parameter estimation 13

4 LDA with Soft Assignment of Descriptors to Words **14**

4.1 LDA with Generative Words 14

4.2 Inference 16

4.3 Parameter estimation 17

5 Dictionary Of Generative Words **20**

5.1 Dynamic Texture 20

5.2 Learning a Dictionary of Dynamic Textures 22

5.3 Dynamic Textures examples 23

6 Experimental Results **24**

6.1 UCSD Dataset 25

6.2 UMN Dataset 27

7 Summary and discussion **30**

Abstract

Automatic processing of video data is essential in order to allow efficient access to large amounts of video content, a crucial point in applications such as video mining and surveillance. In this paper we focus on the problem of identifying interesting parts of the video. Specifically, we seek to identify atypical video events, which are the events a human user is usually looking for. As is common in the literature, we equate atypical events to events of low-probability with respect to a model that describes normal events. We propose to identify atypical events by modeling a corpus of typical video events using the Latent Dirichlet Allocation model. Subsequently, to classify an event as atypical we compute its probability with respect to the learned LDA model. Furthermore, we've extended the LDA model, which works with discrete data, to work with continuous data as in our case of video events. We tested our algorithm on the UCSD ped2 [20] and UMN datasets [1], which were previously used to evaluate anomaly detection algorithms.

Chapter 1

Introduction

1.1 Motivation

As more and more video cameras are placed throughout cities, malls, airports, etc. due to cameras affordability and availability, the sheer amount of generated video footage increases tremendously. This poses a problem as organizations need more manpower to monitor these cameras, but also humans surveying large amounts of footage are prone to missing significant incidents due to lack of attention. For these reasons automatic processing of video data is crucial in processing such large amounts of data.

Some solutions for this problem include video summarization [26] where activities originating from different time periods are displayed simultaneously, categorization, novelty detection, etc. We are specifically interested in the task of novelty detection, or anomaly detection, where one wants to detect unusual events for further processing, for example in order to alert security.

We suggest an automatic, human free process that accomplishes this task by identifying unusual activities in the video. The process consists of extracting the activities from the video, establishing a model that describes regular activities and identifying unusual activities by an inference method.

1.2 Our Approach

We propose a unsupervised approach to learn typical, normal events using a generative model (Latent Dirichlet Allocation[3]) and test it on two datasets, which were previously used to evaluate abnormality detection in Computer Vision algorithms.

Given a collection of training videos containing only normal activities, we decompose the videos to a corpus of events represented as a "Bag-Of-Words". Unlike the traditional representations of "Bag-Of-Words" which assign each feature to the "best" matching word, we soft assign each feature to each word according to a "matching" measurement. The detected activities are then represented using a latent topic model, a paradigm that has already shown promising results [17, 23, 12, 15].

The training stage concludes with learning an LDA model that maximizes the likelihood of the data. Furthermore we extend the LDA model to handle events composed of soft assigned features.

In the test stage, we classify events as abnormal if their posterior probability given the learned model is low, and normal if their posterior probability is high.

1.3 Related Work

Various statistical measurements have been suggested to detect abnormal events in video, for example, Saliency, statistical outliers over space and time [19]. [16] suggested to measure the difference between prior and posterior distributions over the set of all models using Kullback-Liebler (KL) divergence. [4, 7] formulated the problem as the problem of composing the questioned event using spatio-temporal patches [4] or over-complete normal basis [7] extracted from training examples.

We continue the work of [13] who proposed using LDA to model activities represented as a histogram of the object transitions, and identify abnormal activities using Bysian surprise, where the model is updated using the queried activity.

Papers such as [15, 34] use topic modeling to represent the environment by extending the LDA model. Both used low-level features based on optical flow to represent the activities. [15] introduced a third latent layer depicting the category of the activity and exploited the relation between sequential activities by assuming a multinomial distribution over the evolution of sequential categories and used Bayesian saliency to recognize irregular patterns. [34] modeled a cluster of documents by assuming different prior over the topics distribution and classified abnormal activities as activities with low likelihood.

Other works in Computer Vision used and extended the LDA model for scene categorization [12], object discovery in an image [28, 30, 33, 5], visual class hierarchies [29], time based activities discovery in a video [11], human action categorization [23] and tracking individuals in a crowded scenes [27].

Papers as [10, 24, 25] also addressed the problem of assigning a visual descriptor to a single word, as in most "Bag-Of-Words" representations, by describing a descriptor as a mixture probability over words, where each document is described as a histogram of a sum of probabilities (aka responsibility) or some other distance measurement.

The rest of the manuscript is organized as follows: in Chapter 2 we define what is an event and how it is represented using soft assignment. Chapter 3 describes the Latent Dirichlet Allocation model. In Chapter 4 we extend the LDA model to handle soft assignments. Chapter 5 describes the generative dictionary we used and how it is learned in the training phase, while in Chapter 6 we present test results on benchmark datasets.

Chapter 2

Event Representation Using Bag Of Words

2.1 Tracking-Free Pedestrian Events

Trying to classify a single pedestrian's whole movement might be complicated due to various issues: Object tracking is needed to capture the pedestrian's movement throughout the video; Occlusions should be handled to avoid breaking down a single pedestrian's movement to multiple pedestrians; Crowded environments could introduce difficulties in tracking or analyzing the video, i.e creating the actual representation. Also, it is unclear how to handle pedestrian movement that is part normal and part abnormal. For example, a pedestrian walks (normally) along a sidewalk but starts to skip (abnormally) when he leaves the frame. Should it be considered as an abnormality or not?

To avoid these issues we're considering events as spatio-temporal patches, extracted using a "3D sliding window", where each event captures a small part of the video. Therefore an event might contain multiple pedestrians moving in different directions. Using this method we're avoiding using trackers and there's no need to handle occlusions. Also instead of classifying the pedestrian's entire movement, we're classifying a small space-time portion of the video which helps us with the partial abnormal movement problem described above.

Note that the event's size decision presents a trade-off between the resolution of the event, and our ability to properly analyze it. Small event isolate a small part of the video, but if it is too small it might be difficult to understand what happens inside it. On the other hand, using larger events could capture more meaningful data, but might cause anomalies to be obscured by the "normal part" of the event.

2.2 Bag Of (Hard Assigned) Words

A common approach in Computer Vision is to represent videos (and also images) using the Bag-Of-Words approach, originally developed for text analysis: a set of visual features is extracted from the video, using methods such as feature detectors [22, 8], regular grid, random sampling, etc. A dictionary of visual words is constructed from the set of extracted features, for example by vector quantization of the observed features using K-Means. Each visual feature is assigned to the *closest* visual word from the dictionary. Lastly a video is represented by a histogram of feature-to-words assignments.

Although numerous algorithms use the Bag-Of-Words approach, it holds several problems such as:

- Spatial information is ignored - the location of the feature and the spatial relation between the features are disregarded.
- Each feature is *hard* assigned to the *closest* visual word, in the sense that we're disregarding: (1) how close it is to the *closest* word, and (2) how close it is to other words.

Let's elaborate about the 2nd bullet using Fig. 2.1:

1. The Bag-Of-Words approach doesn't handle the ambiguity of visual patches that are at a similar distance to multiple, close visual words. As could be seen by the right patch (x) in Fig. 2.1 - it is not clear as to which of the visual words it should be assigned.

2. Also the Bag-Of-Words assumes that the visual features in the test video/image come from the same distribution as the trained dictionary, i.e. the dictionary can represent each visual feature with a small quantization error. In Fig. 2.1, the left patch in the image will be assigned to the red or green words, but it is clear that neither of them is a good representation of it.

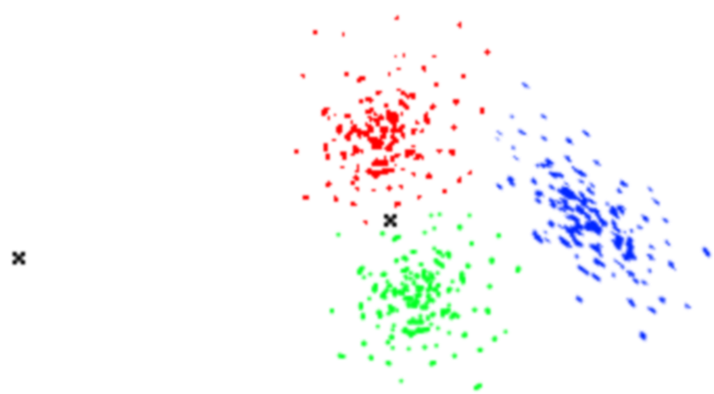


Figure 2.1: Hard assignment example - 2D space displaying a dictionary composed of 3 words learned from training patches and 2 new patches (marked as "x")

In our application, where we learn a dictionary from the training videos and use it to represent patches from the test videos, the 2nd problem will often arise as the test video will contain abnormal events with new, unforeseen patches. As a result the events' abnormality will ofter occur not just because the underline structure of the event but also due to the incompatibility of features to the dictionary.

2.3 Bag Of Soft Assigned Words

To handle the hard assignment problem described in the previous section, we'll represent an event using soft assignment - each feature in the event is associated with each visual word according to their distance, or by some other similarity measurement (see Fig. 2.2)

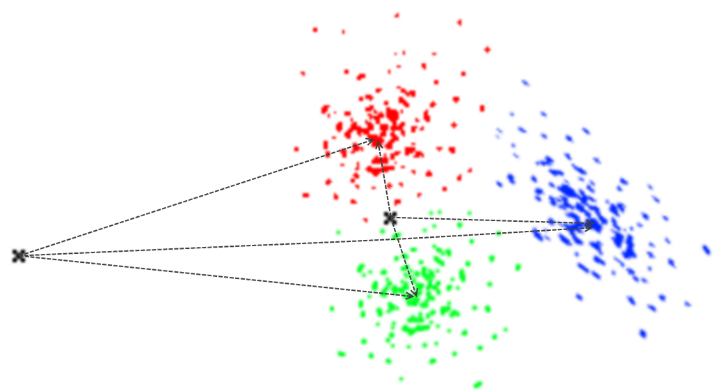


Figure 2.2: Soft assignment example - each feature is soft assigned to each word

In more detail, given a dictionary of generative words (for examples see chapter 5) and a set of videos, each video is decomposed into a set of spatio-temporal events, and each event is further decomposed to smaller spatio-temporal patches (aka *3D-patches*). Finally, an event is represented by the probabilities assigned by each generative word to each 3D-patch (see Fig 2.3).

Previous works that used soft assignment, assigned each descriptor to the k-nearest words with equal weight [32], or by some weighting scheme [18, 31] and constructed a soft assignment histogram, where the i-th bin contains the sum of weights assigned to the i-th word. In Chapter 4 we'll see that our representation of an event is the soft assignment histogram, and also the average scaled covariance matrix of the soft assignments, capturing both the 1st and the 2nd order statistics of the soft assignment of descriptors in the event.

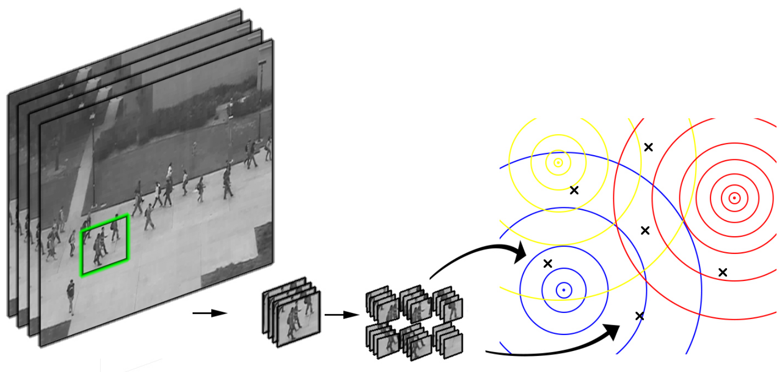


Figure 2.3: A video sequence (left) is divided into video events (marked with a green border). Each event is divided into small spatio-temporal patches, and each patch is represented by a mixture of words

Chapter 3

Modeling of Typical Events Using LDA

3.1 Latent Dirichlet Allocation

Latent Dirichlet Allocation[3] (LDA) model is a generative model originally developed for statistical text analysis. LDA models a corpus of documents represented as a Bag-Of-Words as a mixture of topics. Each topic is defined by a multinomial distribution over words in a pre-defined vocabulary.

LDA models a corpus of documents in an unsupervised manner, meaning it can handle large volumes of data with no human supervision. As a generative model LDA defines a crude posterior probability for documents, enabling to classify normal and abnormal documents. Unlike its predecessor pLSA[14], it doesn't suffer from over-fitting problems.

LDA describes how each document was generated through the following generative process: see fig. 3.1a for a graphical model using plate notation.

1. Choose $N \sim Poisson(\xi)$, the number of words in the document.

2. Choose $\theta \sim Dirichlet(\alpha)$, the topics distribution within the document.

3. For each of the N words w_n, where $n = 1 .. N$:
    (a) Choose a topic $z_n \sim Multinomial(\theta)$
    (b) Choose a word w_n from the multinomial distribution $p(w_n \mid z_n, \beta)$

The LDA model parameters are (1) α: a k-dimensional Dirichlet parameter (k is the number of topics), which defines the distribution of the topic mixture. (2) β: a $k \times V$ (vocabulary length) matrix defining the words distribution for each topic, where each row holds the words' multinomial distribution for each topic.

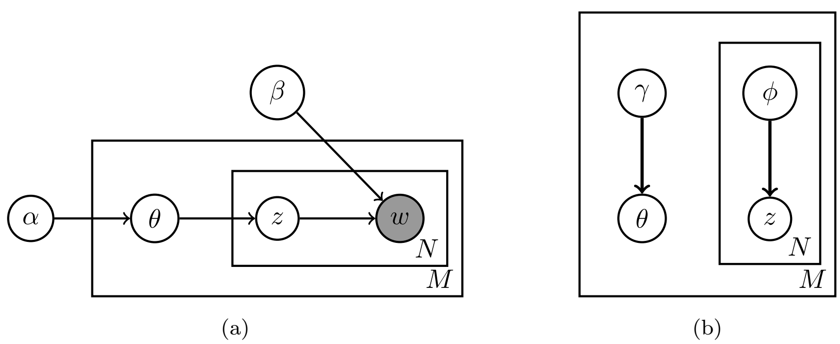


Figure 3.1: (a) Graphical model representation of LDA using plate notation. (b) Graphical model representation of the variational distribution used to approximate the posterior in LDA.

3.2 Inference

Given the parameters α and β, the joint distribution of a topic mixture θ, a set of N topics z, and a set of N words w is given by:

$$p(\theta, \mathbf{w}, \mathbf{z} \mid \alpha, \beta) = p(\theta \mid \alpha) \prod_{n=1}^{N} p(z_n \mid \theta) p(w_n \mid z_n, \beta) \tag{3.1}$$

To use the LDA model we need to compute the posterior distribution of the hidden variables given a document w and the model parameters α and β:

$$p(\theta, \mathbf{z} \mid \mathbf{w}, \alpha, \beta) = \frac{p(\theta, \mathbf{w}, \mathbf{z} \mid \alpha, \beta)}{p(\mathbf{w} \mid \alpha, \beta)} \tag{3.2}$$

Unfortunately this distribution is intractable due to the coupling between θ and β. A wide variety of approximate inference algorithms are used to compute an approximation for this posterior probability, including Gibbs sampling, Laplace approximation, Markov chain Monte Carlo and variational approximation, which we will use.

The basic idea of variational approximation is to use a simplified model which induces a lower bound on the questioned function, in our case the posterior probability. As the variational model becomes more similar to the original model the lower bound gets tighter.

Fig. 3.1b displays a graphical representation of the simplified variational model. γ is a document's k-value approximation of the Dirichlet parameter α. For each word n, ϕ_n replaces θ such that the topic distribution of word n is multinomial with ϕ_n.

The new variational model defines the variational distribution:

$$q(\theta, \mathbf{z} \mid \gamma, \phi) = q(\theta \mid \gamma) \prod_{n=1}^{N} q(z_n \mid \phi_n) \tag{3.3}$$

Blei [3] showed that when using Jensen's inequality the dependencies between the models follows:

$$\log p(w \mid \alpha, \beta) = L(\gamma, \phi; \alpha, \beta) + D_{kl}(q(\theta, z \mid \gamma, \phi) \mid\mid p(\theta, z \mid w, \alpha, \beta)) \tag{3.4}$$

Where $L(\gamma, \phi; \alpha, \beta)$ is defined by:

$$\begin{aligned} L(\gamma, \phi; \alpha, \beta) &= E_q[\log p(\theta \mid \alpha)] + E_q[\log p(z \mid \theta) \\ &+ E_q[\log p(w \mid z, \beta)] \\ &- E_q[\log q(\theta)] - E_q[\log q(z)] \end{aligned} \tag{3.5}$$

As we can see $L(\gamma, \phi; \alpha, \beta)$ is indeed a lower bound on the log likelihood as KL-divergence is always non-negative. Also as the distance between the two probabilities decreases (the models become more "similar") the lower bound gets tighter.

Therefore in order to tighten the lower bound we need to minimize the KL-divergence, resulting in the following optimization problem:

$$(\gamma^*, \theta^*) = \arg\min_{\gamma,\theta} D_{kl}(q(\theta, z \mid \gamma, \phi) \mid\mid p(\theta, z \mid w, \alpha, \beta)) \tag{3.6}$$

By computing the derivatives of the KL-divergence with respect to γ and ϕ and setting them to zero, the following updated equations are obtained:

$$\phi_{ni} \propto \beta_{iw_n} \exp(E_q[\log \theta_i \mid \gamma]) = \beta_{iw_n} \exp(\Psi(\gamma_i) - \Psi(\sum_{j=1}^{k} \gamma_j)) \tag{3.7}$$

$$\gamma_{ni} = \alpha_i + \sum_{n=1}^{N} \phi_{ni} \tag{3.8}$$

Using these equations the optimization problem can be solved via an iterative fixed-point method.

We will note that due to exchangeability, ϕ_n for a given document is the same for all word locations n where the same word w_a is observed in the original bag of words model. Therefore we can define a vector of length V $\hat{\phi}_a$, where $\hat{\phi}_a = \phi_n$ such that $w_n = a$ (word w_a is observed in location n), resulting in the following update rules:

$$\hat{\phi}_{ai} \propto \beta_{ia} \exp(\Psi(\gamma_i) - \Psi(\sum_{j=1}^{k} \gamma_j)) \tag{3.9}$$

$$\gamma_{ni} = \alpha_i + \sum_{a=1}^{V} \hat{\phi}_{ai} cnt(a) \tag{3.10}$$

where $cnt(a)$ is the number of times word 'a' appeared in the document.

3.3 Parameter estimation

In order to estimate the model parameters, we need to find α and β that maximize the marginal log likelihood of the data $\{w_1, ..., w_M\}$:

$$l(\alpha, \beta) = \sum_{d=1}^{M} \log p(w_d \mid \alpha, \beta) \tag{3.11}$$

Blei [3] suggested an alternating variational EM procedure to estimate the model parameters:

1. E-Step: For each document, find the optimizing values of the variational parameters γ^*, ϕ^*
2. M-Step: Maximize the resulting lower bound on the log likelihood of the entire corpus with respect to the model parameters α and β.

The procedure iterates between maximizing the sum of lower bounds with respect to the variational parameters γ^*, ϕ^* (E-step) and maximizing the sum of lower bounds with respect to the model parameters α and β (M-step), until the lower bound on the log likelihood converges.

To maximize with respect to β, $L(\gamma, \phi; \alpha, \beta)$ is differentiated and a Lagrange multiplier is added, resulting in the following update rules:

$$\beta_{ij} \propto \sum_{d=1}^{M} \sum_{n=1}^{N_d} \phi_{dni} w_{dn}^{j} = \sum_{d=1}^{M} \sum_{a=1}^{V} \hat{\phi}_{dai} \; cnt_d(a) \tag{3.12}$$

Updating the Dirichlet parameter α can be implemented using an efficient Newton-Raphson method in which the Hessian is inverted in linear time.

Chapter 4

LDA with Soft Assignment of Descriptors to Words

In the previous chapter we've described the LDA model, which assumes that the words composing the documents in the corpus come from a discrete, finite dictionary. In the following chapter we will extend the LDA to model a corpus of documents composed of continuous descriptors, whose distribution we model as a mixture over a set of discrete generative words, and each generative word assigns a probability to every descriptor. Therefore we soft assign each descriptor to each generative word with respect to its probability.

We assume that the dictionary of generative words is learnt beforehand during the pre-processing of the training data and we only need to handle topic modeling. In Chapter 5 we give example of generative dictionary and describe the dictionary learning process.

4.1 LDA with Generative Words

Our extension to the LDA model [3] lies in the last part of the generative process assumed by the LDA; instead of choosing a discrete word our extension chooses a generative word, and from that word a descriptor is generated. Therefore, the extended model assumes the following generative process for each document: (see Fig.4.1 for a graphical model)

1. Choose $N \sim Poisson(\xi)$, the number of words in the document.
2. Choose $\theta \sim Dirichlet(\alpha)$, the topics distribution within the document.
3. For each of the N descriptors x_n, where $n = 1 .. N$:
   (a) Choose a topic $z_n \sim Multinomial(\theta)$
   (b) Choose a **generative word** w_n from the multinomial distribution $p(w_n \mid z_n, \beta)$
   (c) Choose descriptor x_n from the conditional distribution $p(x_n \mid w_n)$ defined by the generative word model of w_n

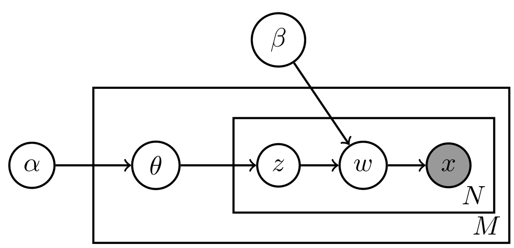


Figure 4.1: (a) Graphical model representation of LDA with Generative Words

Consequently, given the parameters α and β, the joint distribution of the hidden variables (θ, set of z's, set of w's) and the observable variables (set of x's) is given by:

$$p(\theta, z, w, x \mid \alpha, \beta) = p(\theta \mid \alpha) \prod_{n=1}^{N} p(z_n \mid \theta)\ p(w_n \mid z_n, \beta)\ p(x_n \mid w_n) \tag{4.1}$$

4.2 Inference

As in the original LDA model, in order to calculate the posterior probability of a document x we marginalize over the hidden variables:

$$p(x \mid \alpha, \beta) = \int_{\theta} \sum_{z} \sum_{w} p(\theta, z, w, x \mid \alpha, \beta) \, \mathrm{d}\theta \tag{4.2}$$

This term is also intractable. Therefore, we follow Blei [3] variational inference as described in the previous chapter to obtain a lower bound on the log likelihood of the document x:

$$p(x \mid \alpha, \beta) \geq L(\gamma, \phi; \alpha, \beta) \tag{4.3}$$

where $L(\gamma, \phi; \alpha, \beta)$ is defined as:

$$\begin{aligned} L(\gamma, \phi; \alpha, \beta) &= E_q[\log p(\theta \mid \alpha)] + E_q[\log p(z \mid \theta)] \\ &+ E_q[\log p(x \mid z, \beta)] \\ &- E_q[\log q(\theta)] - E_q[\log q(z)] \end{aligned} \tag{4.4}$$

The resulting lower bound is almost identical to the original lower bound, except for the 3rd term:

$$E_q[\log p(x \mid z, \beta)] = \sum_{n=1}^{N} \sum_{i=1}^{k} \phi_{ni} \log[\sum_{j=1}^{V} \beta_{ij} f_{nj}] \tag{4.5}$$

where $f_{nj} = p(x_n \mid w_j)$ is defined by the generative dictionary by soft assigning a feature to each generative word.

By deriving the lower bound with respect to γ and ϕ we get the following update rules:

$$\phi_{ni} \propto (\sum_{j=1}^{V} \beta_{ij} f_{nj}) \exp(\Psi(\gamma_i) - \Psi(\sum_{j=1}^{k} \gamma_j)) \tag{4.6}$$

$$\gamma_i = \alpha_i + \sum_{n=1}^{N} \phi_{ni} \tag{4.7}$$

where Ψ is the digamma function.

In the extended model where the (generative) word is now a hidden variable we still define $\hat{\phi}$ in the same manner, $\hat{\phi}_a = \phi_n$ s.t. $w_n = a$. The update rule for

$\hat{\phi}_a$ remains:

$$\begin{aligned} \hat{\phi}_{ai} &\propto \beta_{ia} \exp(\Psi(\gamma_i) - \Psi(\sum_{j=1}^{k} \gamma_j)) && (4.8) \\ \hat{\phi}_{ai} &= \frac{\beta_{ia}}{Q_a} \exp(\Psi(\gamma_i) - \Psi(\sum_{j=1}^{k} \gamma_j)) && (4.9) \end{aligned}$$

where $Q_a = \sum_{i=1}^{k} \beta_{ia} \exp(\Psi(\gamma_i) - \Psi(\sum_{j=1}^{k} \gamma_j))$.

A crude approximation allows us to derive very similar update rules for the variational parameters:

$$\phi_{ni} \approx \sum_{a=1}^{V} \hat{\phi}_{ai} \frac{f_{na}}{F_n} \tag{4.10}$$

Where $F_n = \sum_{a=1}^{V} f_{na}$. This normalized vector approximates the expression in 4.7, and is true when β_{ij} is the same for all topics j. Now

$$\begin{aligned} \gamma_i &= \alpha_i + \sum_{n=1}^{N} \phi_{ni} \approx \alpha_i + \sum_{n=1}^{N} \sum_{a=1}^{V} \hat{\phi}_{ai} \frac{f_{na}}{F_n} \\ \gamma_i &\approx \alpha_i + \sum_{a=1}^{V} \hat{\phi}_{ai} PseudoCnt(a) && (4.11) \end{aligned}$$

where $PseudoCnt(a) = \sum_{n=1}^{N} \frac{f_{na}}{F_n}$ denotes the sum of normalized probabilities.

4.3 Parameter estimation

As before, in order to estimate the model parameters α and β we maximize the sum of the documents' lower bound $L_x(\gamma, \phi; \alpha, \beta)$ with respect to α and β.

As we saw earlier, the only term in the lower bound $L(\gamma, \phi; \alpha, \beta)$ which depends on α (1st term in Eq. 4.4) has not changed from the original lower bound. Therefore, when deriving the lower bound with respect to alpha, we get the same update rule and maximizing the lower bound with respect to α could be done in the same manner as in the previous section, using Newton-Raphson algorithm.

Maximizing the sum of the lower bound with respect to β is equivalent to maximizing the sum of the 3rd term - $E_q[\log p(x \mid z, \beta)]$ (the only term depending on β) subject to the constraint that the columns of β sum to 1.

Using the concavity of the log function we get a less tighter lower bound that is easily maximized:

$$\log[\sum_{j=1}^{V} \beta_{ij} f_{xnj}] = \log[(\sum_{l=1}^{V} f_{xnl})(\sum_{j=1}^{V} \beta_{ij} \frac{f_{xnj}}{\sum_{l=1}^{V} f_{xnl}})] \geq \log(F_{xn}) + \sum_{j=1}^{V} \frac{f_{xnj}}{F_{xn}} \log \beta_{ij} \tag{4.12}$$

where $F_{xn} = \sum_{l=1}^{V} f_{xnl}$, and we get the lower bound:

$$\begin{aligned} \sum_{x \in Corpus} E_q[\log p(x \mid z, \beta)] &= \sum_{x \in Corpus} \sum_{n=1}^{N_x} \sum_{i=1}^{k} \phi_{xni} \log[\sum_{j=1}^{V} \beta_{ij} f_{xnj}] \\ &\geq \sum_{x \in Corpus} \sum_{n=1}^{N_x} \sum_{i=1}^{k} \phi_{xni} [\log(F_{xn}) + \sum_{j=1}^{V} \frac{f_{xnj}}{F_{xn}} \log \beta_{ij}] \end{aligned} \tag{4.13}$$

We maximize this function with respect to β under the normalization constraint by differentiating with respect to β and setting the derivative to zero, resulting in the following update rule:

$$\begin{aligned} \beta_{ij} &\propto \sum_{x \in Corpus} \sum_{n=1}^{N_x} \phi_{xni} \frac{f_{xnj}}{F_{xn}} \\ &= \sum_{x \in Corpus} \sum_{n=1}^{N_x} \sum_{a=1}^{V} \hat{\phi}_{xai} \frac{f_{xna}}{F_{xn}} \frac{f_{xnj}}{F_{xn}} \\ &= \sum_{x \in Corpus} \sum_{a=1}^{V} \hat{\phi}_{xai} \sum_{n=1}^{N_x} \frac{f_{xna}}{F_{xn}} \frac{f_{xnj}}{F_{xn}} \end{aligned} \tag{4.14}$$

where Eq. 4.7 is used in the transition from the 1st to the 2nd lines.

By denoting A_x the average covariance matrix for the probability vectors $\frac{f_{xn}}{F_{dn}}$: $A^x = \sum_{n=1}^{N_x} \frac{f_{xn}}{F_{xn}} \cdot \frac{f_{xn}^T}{F_{xn}}$, 4.14 could be written as:

$$\beta_{ij} \propto \sum_{x \in Corpus} \sum_{a=1}^{V} \hat{\phi}_{xai} A_{aj}^x \tag{4.15}$$

We can see from the update rules that there is no need to save all of the soft assignments of each feature to each word f_{nj} individually, but just the k-vector PseudoCnt (sum of assignments to each word) and the $V \times V$ average covariance matrix A^x, for inference (only the former) and estimating the model parameters (both).

Chapter 5

Dictionary Of Generative Words

In the previous chapter, the extended LDA model assumed a dictionary of generative words is learnt before hand, where each word induces a likelihood function over the continues descriptors space, and the model soft assigns each descriptor to each generative word according to the likelihood function.

In the following chapter we will give an example of a generative words dictionary and describe methods for learning it.

5.1 Dynamic Texture

Dynamic Texture[9] (DT) is a video generative model, capturing both the appearance of the video (textures) and the dynamic components in it.
A DT consists of a random process containing an *observed* variable y_t, which encodes the video frame at a time t, and a *hidden state* variable x_t, which encodes the evolution of the video over time. The state and observed variables are related through the following linear dynamical system equations: (see Fig. 5.1 for a graphical model representation)

$$
\begin{cases}
x_{t+1} = Ax_t + v_t \\
y_t = Cx_t + w_t
\end{cases}
\tag{5.1}
$$

where $y_t \in \mathcal{R}^m$ is the observed state variable, $x_t \in \mathcal{R}^n$ is the hidden state variable, $A \in \mathcal{R}^{n\times n}$ is the state translation matrix, $C \in \mathcal{R}^{n\times m}$ is the 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 respectively, and the initial state x_1 which is usually distributed as $x_1 \sim N(\mu, S)$.

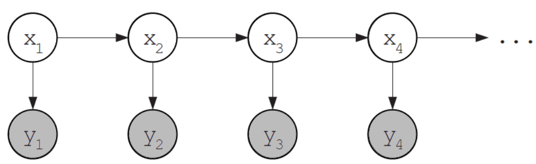


Figure 5.1: Graphical model representation of the Dynamic Texture model

5.2 Learning a Dictionary of Dynamic Textures

Given a set of 3D-Patches collected from the training videos, we want to learn a dictionary of words that maximize the likelihood of the data. We accomplish this task using the following "k-means" algorithm:

Algorithm 1 K-Means

Input: V - Dictionary size (number of DTs) and a Set of 3D-Patches

Initialize: Pick V 3D-patches.

For each patch, learn the DT that maximizes its likelihood.

repeat

Assignment Step:

Assign each 3D-patch to the DT that maximizes its likelihood.

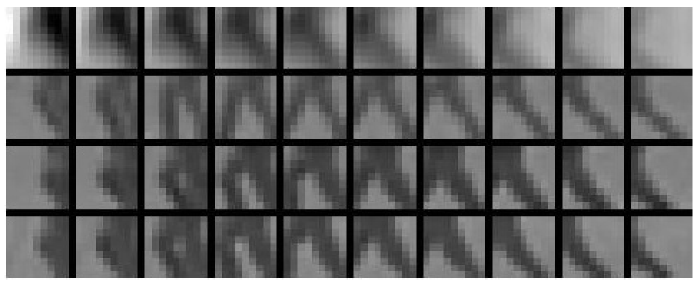
Update Step:

Recalculate each DT according to the 3D-patches assigned to it.

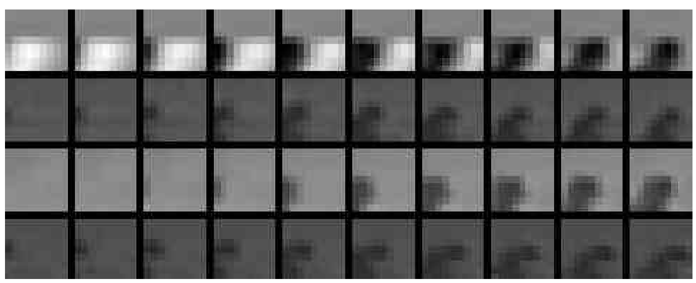
until convergence

5.3 Dynamic Textures examples

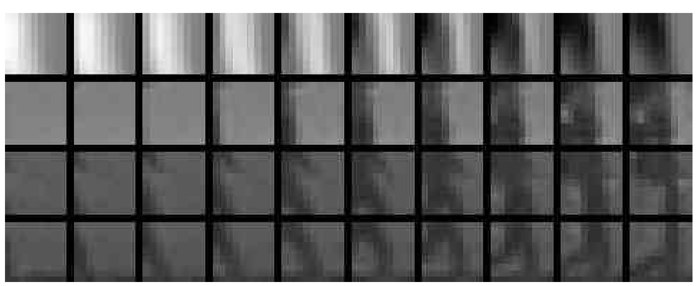
Fig 5.2 displays examples of learned DTs using the described "K-means" algorithm. We can see from these examples that the Dynamic Textures indeed capture both the dynamics (movement direction) and the texture (the object).



(a) A leg moving to the left.



(b) A head moving to the right



(c) A torso moving to the right.

Figure 5.2: Illustration of three DT words (frames are ordered from left to right). In each DT: First row: a synthetic 3D-patch generated from the DT. Subsequent rows: three 3D-patches assigned to the DT by the "k-means" algorithm.

Chapter 6

Experimental Results

To evaluate our algorithm we used the UCSD Ped2 dataset [20] and the UMN dataset [1]. Both were previously used to test abnormality detection in Computer Vision algorithms [19, 7, 35, 21].

Both datasets contain scenes with walking pedestrians. In the UCSD dataset the abnormal events include bicycle and skateboard riders and a cart (see Fig 6.1 for examples of detections) and in the UMN dataset the abnormal events include frames where all of the pedestrians escape out of the scene (see Fig 6.3 for normal and abnormal example frames). Hence, the UCSD dataset contains abnormalities at *specific regions* in the frame and the UMN contains entire *abnormal frames*.

In both datasets each video was split into events of size $24 \times 24 \times 21$ for the UCSD dataset (21 is the temporal length and the frame size is 360×240) and $320 \times 240 \times 10$ for the UMN dataset (the entire frame). Each event is decomposed into smaller 3D-patches of size $9 \times 9 \times 10$ (UCSD - $4 \times 4 \times 3$ 3D-patches in each event; UMN - $60 \times 44 \times 1$). To improve performance we filtered out events and 3D-patches with no movement. A dictionary was computed from the collection of 3D-patches, as described in chapter 5.

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 training set, searching for the best size over a feasible range. The optimal dictionary size was 75 words, with the hidden state size set to 10 as described in chapter 5.

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 some of the lower assignment values to zero, typically those below the average assignment [6]. Here it was necessary to use a higher threshold and set up to 95% of the lower assignments to zero, 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 2.3. Finally, using events from the training video an extended LDA model was learnt as described in chapter 4, and subsequently used to estimate the likelihood of each event in the test videos. Using the event's likelihood, we identify events whose likelihood is below a certain threshold as abnormal events.

Following [19], we tested our method using frame-level (both datasets) and pixel-level (only UCSD dataset) measurements, where in the former a detection in a frame is successful regardless of the abnormality location within the frame, and in the latter a detection is successful if at least 40% of the truly anomalous pixels are detected.

6.1 UCSD Dataset

Fig. 6.1 displays frames with abnormal events detected by our algorithm.

Figure 6.1: Examples of abnormal events in the UCSD ped2 dataset.

We tested our algorithm using hard assignment to words, corresponding to the original LDA model and soft assignment to words (our method). Fig 6.2 displays results as ROC curves and frame level EER are tabulated in table 6.1. It is clear that soft assignment achieved top performance, better than alternative methods and much better than hard assignment.

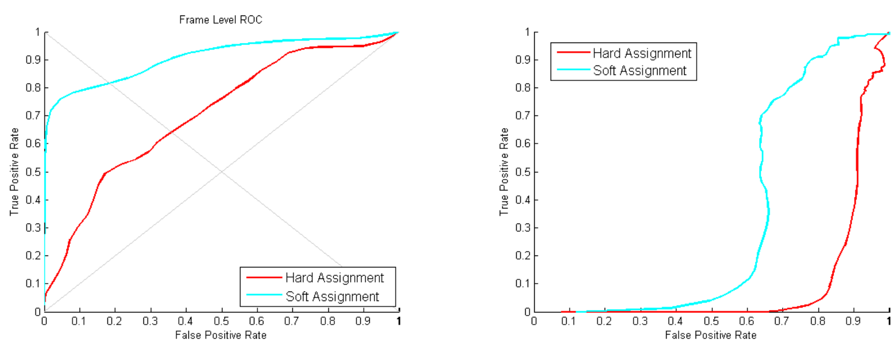


Figure 6.2: ROC curves for frame level analysis (left) and pixel level (right).

method	EER
SF [19]	42%
MPPCA [19]	30%
SF-MPPCA [19]	36%
Adam [2]	42%
MDT [19]	25%
Soft assignment	18.5%
Hard assignment	35%

Table 6.1: Frame level EER values for different methods.

6.2 UMN Dataset

Fig. 6.3 displays normal frames and abnormal frames detected by our algorithm.

Figure 6.3: Examples for frames from the 3 scenes in the UMN dataset. Top row: normal frames. Bottom row: abnormal frames

Here we tested only frame-level detection as abnormalities are represented as frames with all of the people escaping out of the scene. Fig 6.4 displays results as

ROC curves and Table 6.2 provides quantitative comparisons to other state-of-the-art algorithms. Except for scene 2, our algorithm got results similar to those obtained by state-of-the-art algorithms. In scene 2 most of the normal frames consist of pedestrians who stand still. Since our representation of *Dynamic Textures* isn't capable of modeling the lack of movement, it attempts to model the noise in the videos.

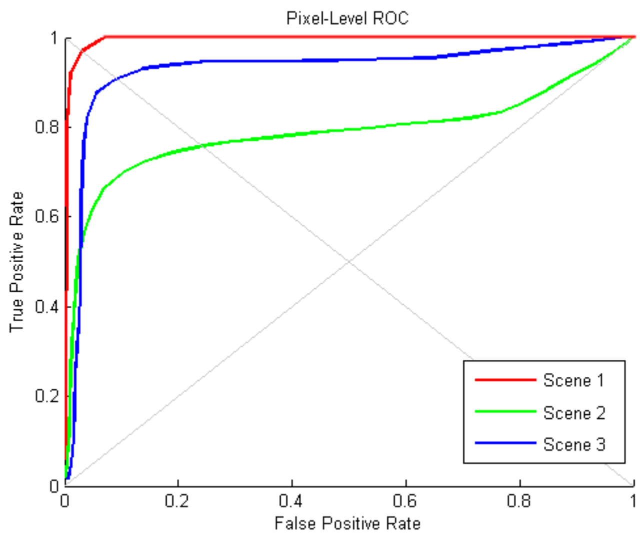


Figure 6.4: ROC curves for UMN dataset.

method	Area under ROC
Chaotic Invariants [35]	0.99
Social Force [21]	0.96
Optical ow [21]	0.84
Sparse Reconstruction Cost [7] - Scene 1	0.99
Sparse Reconstruction Cost [7] - Scene 2	0.97
Sparse Reconstruction Cost [7] - Scene 3	0.96
Soft assignment - Scene 1	0.994
Soft assignment - Scene 2	0.786
Soft assignment - Scene 3	0.927

Table 6.2: Frame level AUC values for different methods.

Chapter 7

Summary and discussion

Latent Dirichlet Allocation has proven very effective in modeling text, images, videos and other multi-media documents. We extended this model to allow for a richer representation, in order to deal 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 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.

We compared our algorithm to state-of-the-art algorithms using real world data of walking pedestrians and got similar results. We acknowledge that we did not get good results for UCSD ped1 dataset [20] and the 2nd scene of the UMN dataset [1], mainly because the Dynamic Textures weren't able to learn some the textures causing the model to identify pedestrians with white clothes as abnormalities. Also in the 2nd scene of the UMN dataset [1] the training part of the video mostly contained standing people, causing the Dynamic Textures to be over-sensitive to the noise, and therefore unable to model the lack of movement. We tried various methods of normalization (such as Whitening) but they did not improve the results.

Bibliography

[1] Unusual crowd activity dataset of university of minnesota. http://mha.cs.umn.edu/movies/crowdactivity-all.avi.

[2] Amit Adam, Ehud Rivlin, Ilan Shimshoni, and Daviv Reinitz. Robust real-time unusual event detection using multiple fixed-location monitors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(3):555–560, 2008.

[3] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.

[4] Oren Boiman and Michal Irani. Detecting irregularities in images and in video. *International Journal of Computer Vision*, 74(1):17–31, 2007.

[5] Liangliang Cao and Li Fei-Fei. Spatially coherent latent topic model for concurrent segmentation and classification of objects and scenes. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[6] Adam Coates, Honglak Lee, and Andrew Y Ng. An analysis of single-layer networks in unsupervised feature learning. *Ann Arbor*, 1001:48109, 2010.

[7] Yang Cong, Junsong Yuan, and Ji Liu. Sparse reconstruction cost for abnormal event detection. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3449–3456. IEEE, 2011.

[8] Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, page 22, 2004.

[9] Gianfranco Doretto, Alessandro Chiuso, Ying Nian Wu, and Stefano Soatto. Dynamic texturbes. *International Journal of Computer Vision*, 51(2):91–109, 2003.

[10] Jason Farquhar, Sandor Szedmak, Hongying Meng, and John Shawe-Taylor. Improving" bag-of-keypoints" image categorisation: Generative models and pdf-kernels. 2005.

[11] Tanveer A Faruquie, Prem Kumar Kalra, and Subhashis Banerjee. Time based activity inference using latent dirichlet allocation. In *BMVC*, pages 1–10, 2009.

[12] Li Fei-Fei and Pietro Perona. A bayesian hierarchical model for learning natural scene categories. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 524–531. IEEE, 2005.

[13] Avishai Hendel, Daphna Weinshall, and Shmuel Peleg. *Identifying surprising events in videos using bayesian topic models.* Springer, 2011.

[14] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. ACM, 1999.

[15] Timothy Hospedales, Shaogang Gong, and Tao Xiang. A markov clustering topic model for mining behaviour in video. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1165–1172. IEEE, 2009.

[16] Laurent Itti and Pierre Baldi. A principled approach to detecting surprising events in video. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 631–637. IEEE, 2005.

[17] Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. *An introduction to variational methods for graphical models*. Springer, 1998.

[18] Piotr Koniusz and Krystian Mikolajczyk. Soft assignment of visual words as linear coordinate coding and optimisation of its reconstruction error. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 2413–2416. IEEE, 2011.

[19] Vijay Mahadevan, Weixin Li, Viral Bhalodia, and Nuno Vasconcelos. Anomaly detection in crowded scenes. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1975–1981. IEEE, 2010.

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

[21] Ramin Mehran, Alexis Oyama, and Mubarak Shah. Abnormal crowd behavior detection using social force model. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 935–942. IEEE, 2009.

[22] Krystian Mikolajczyk and Cordelia Schmid. An affine invariant interest point detector. In *Computer VisionECCV 2002*, pages 128–142. Springer, 2002.

[23] Juan Carlos Niebles, Hongcheng Wang, and Li Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *International Journal of Computer Vision*, 79(3):299–318, 2008.

[24] Florent Perronnin, Christopher Dance, Gabriela Csurka, and Marco Bressan. Adapted vocabularies for generic visual categorization. In *Computer Vision–ECCV 2006*, pages 464–475. Springer, 2006.

[25] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[26] Yael Pritch, Alex Rav-Acha, and Shmuel Peleg. Nonchronological video synopsis and indexing. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(11):1971–1984, 2008.

[27] Mikel Rodriguez, Saad Ali, and Takeo Kanade. Tracking in unstructured crowded scenes. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1389–1396. IEEE, 2009.

[28] Josef Sivic, Bryan C Russell, Alexei A Efros, Andrew Zisserman, and William T Freeman. Discovering objects and their location in images. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 370–377. IEEE, 2005.

[29] Josef Sivic, Bryan C Russell, Andrew Zisserman, William T Freeman, and Alexei A Efros. Unsupervised discovery of visual object class hierarchies. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[30] Tinne Tuytelaars, Christoph H Lampert, Matthew B Blaschko, and Wray Buntine. Unsupervised object discovery: A comparison. *International Journal of Computer Vision*, 88(2):284–302, 2010.

[31] Jan C van Gemert, Jan-Mark Geusebroek, Cor J Veenman, and Arnold WM Smeulders. Kernel codebooks for scene categorization. In *Computer Vision–ECCV 2008*, pages 696–709. Springer, 2008.

[32] Jan C van Gemert, Cor J Veenman, Arnold WM Smeulders, and J-M Geusebroek. Visual word ambiguity. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(7):1271–1283, 2010.

[33] Xiaogang Wang and Eric Grimson. Spatial latent dirichlet allocation. In *Advances in Neural Information Processing Systems*, pages 1577–1584, 2007.

[34] Xiaogang Wang, Xiaoxu Ma, and Eric Grimson. Unsupervised activity perception by hierarchical bayesian models. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[35] Shandong Wu, Brian E Moore, and Mubarak Shah. Chaotic invariants of lagrangian particle trajectories for anomaly detection in crowded scenes. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2054–2060. IEEE, 2010.