University of Trento

Doctoral Thesis

Understanding Visual Information: from Unsupervised Discovery to Minimal Effort Domain Adaptation

Author:
Gloria Zen

Supervisor:
Dr. Nicu Sebe

A thesis submitted in fulfilment of the requirements
for the degree of Doctor of Philosophy

in the

International Doctorate School in Information and Communication Technologies
Department of Information Engineering and Computer Science
Multimedia and Human Understanding Group (MHUG)

April 2015

"If I have seen further it is by standing on the shoulders of giants."

(Bernard of Chartres)

Isaac Newton

Abstract

Visual data interpretation is a fascinating problem which has received an increasing attention in the last decades. Reasons for this growing trend can be found within multiple interconnected factors, such as the exponential growth of visual data (*e.g.* images and videos) availability, the consequent demand for an automatic way to interpret these data and the increase of computational power. In a *supervised machine learning* approach, a large effort within the research community has been devoted to the collection of *training samples* to be provided to the learning system, resulting in the generation of very large scale datasets. This has lead to remarkable performance advances in tasks such as scene recognition or object detection, however, at a considerable high cost in terms of human labeling effort. In light of the *labeling cost* issue, together with the *dataset bias* one, another significant research direction was headed towards developing methods for learning without or with a limited amount of training data, by leveraging instead on data properties like intrinsic redundancy, time constancy or commonalities shared among different domains. Our work is in line with this last type of approach. In particular, by covering different case scenarios - from dynamic crowded scenes to facial expression analysis - we propose a novel approach to overcome some of the state-of-the-art limitations. Based on the renowned bag of words (BoW) approach, we propose a novel method which achieves higher performances in tasks such as learning typical patterns of behaviors and anomalies discovery from complex scenes, by considering the similarity among visual words in the learning phase. We also show that including sparsity constraints can help dealing with noise which is intrinsic to low level cues extracted from complex dynamic scenes. Facing the so called dataset bias issue, we propose a novel method for adapting a classifier to a new unseen target user without the need of acquiring additional labeled samples. We prove the effectiveness of this method in the context of facial expression analysis showing that our method achieves higher or comparable performance to the state of the art, at a drastically reduced time cost.

Acknowledgements

When I finally made the decision to start a PhD I felt it was the right choice. In these years, I've always been accompanied by the same feeling, during the more joyful moments as well as during the tougher ones. Of course, this journey would have not been so extraordinary without the people that contributed to make it so. My first and deeper thanks go to my supervisor Nicu. I wouldn't have literally gone that far without his support. He gave me his trust when taking my decisions as well as the most valuable feedbacks when I needed them. His commitment and enthusiasm at work is an example to me and working within the MHUG group under his guidance was one of the best thing that could have ever happened to me. A huge thought of gratitude also goes to Elisa, my Virgilian guide through the research world, my colleague and friend, she is a source of inspiration both at work and in life. Also, my PhD experience would have not been so meaningful without the wise guidance and support of my internship supervisors Ashish at MSR, José and Adrien at XRCE and Alex at Yahoo!Labs, to whom I am deeply thankful. Living far away from home, family and friends during these internship periods was always rewarded by the invaluable joy of gaining new knowledge and exploring new interesting research directions, together with the possibility to grow in a very stimulating research environment. "*Happiness is only real when shared*" and so are moments at work to me. I want to thank the people I've been collaborating with in these years for the very thoughtful and inspiring discussions, support and friendship, since the very beginning of time: Flora and Virginia, Emilio and Arnoud, Chiara, Stefano and Oswald, Bruno, Francesco, Michele, Paul and Claudio, Enver, Jacopo, María, Sinan, Vika, Jasper, Julian, Paolo R, Ram, Radu, Duba and Alexandra, Kevin, Yan, Negar, Moji, Andreza and all the other mhuggers and disi people. Manuel, Francesca and Andrea for the logistic support and beyond. Professor Ben, Hendrina, Pieter and Stéfan, *i.e.* the most energetic team ever, it was such a pleasure to have you as visitors. Least but not last, the vision team at Yahoo!Labs, namely Jordi, Yale and Paloma: my staying in NY wouldn't have been so special without you. For anywhere I've been staying, I am so grateful to the people that made that place feel like home to me. In primis, the marvelous crew of Via Veneto 144, Stefano, Elena, Luca, Patricia and Ada. My dearest friends, far from the eyes but not from the heart: Silvia, Monica, Anna and Vanda. Aranha, Pisca, Habib, Niki, Stefy, Johnny, Ingrid, Sorella and all the São Salomonicos from the capoeira family: *que o caminho de São Salomão seja sempre guiado pelas estrelas mais brilhantes e pelo mais puro axé.* Matteo and the Westfalia adventures, Silvietta and the Trento-Napoli railways, Donata and Leo, Katya and Andrea, Sara Zo, Matteo, Alessio and Silvia, Christian and Mariano, Elena P, Valentina and Meer, the climbing people and the vie ferrate in Trento and Grenoble, Le Hazard, La Bobine and

Parc Mistral, Jeannine, Michele Ben, Zeynep, Dominik, Diana, Gufri and the XRCE people, itinerari di musica popolare, Daniele and Fan Chaabi, for bringing the loved Mediterranean breeze to the Alpine Lands, Bruno CSO, Ai Castelli Romani, Chinaski, Cafe de La Paix, Bookique, Malombra, H-Demia for all the food, spritz, concerts and post-aperitivi, Piya, Bob, Greg and the Seamonster Lounge, the Turkish family Ertac, Zeynep and Nublu, Paloma's home movie theater, my bikes engines of happiness, Yoga To The People Williamsburg, my dear Luisina y el mate. Finally, my family, Mum and Dad, David and Roberta, Flora, for the endless love and support.

Contents

Publications

This thesis consists of the following publications:

- Chapter 2
  - **G Zen** and E Ricci. "Earth Mover's Prototypes: a Convex Learning Approach for Discovering Activity Patterns in Dynamic Scenes". *International Conference on Computer Vision (CVPR)*, 2011 [1]
  - **G Zen**, E Ricci, S Messelodi and N Sebe. "Sorting atomic activities for discovering spatio-temporal patterns in dynamic scenes". *International Conference on Image Analysis and Processing (ICIAP)*, 2011 [2]
  - E Ricci, **G Zen**, N Sebe and S Messelodi. "A Prototype Learning Framework Using EMD: Application to Complex Scenes Analysis". *IEEE Transactions on Pattern Analysis and Machine Intelligence, 35 (3), 513-526*, 2013 [3]
- Chapter 3
  - **G Zen**, E Ricci and N Sebe. "Exploiting sparse representations for robust analysis of noisy complex video scenes". *European Conference on Computer Vision (ECCV)*, 2012 [4]
  - **G Zen**, J Krumm, N Sebe, E Horvitz, A Kapoor. "Nobody likes Mondays: foreground detection and behavioral patterns analysis in complex urban scenes". *ACM/IEEE international workshop on Analysis and Retrieval of Tracked Events and Motion in Imagery Stream (ARTEMIS)*, 2013 [5]
  - **G Zen**, E Ricci and N Sebe. "Simultaneous Ground Metric Learning and Matrix Factorization with Earth Mover's Distance". *IEEE International Conference on Pattern Recognition (ICPR)*, 2014 [6]
- Chapter 4
  - E Sangineto*, **G Zen***, E Ricci and N Sebe. "We are not All Equal: Personalizing Models for Facial Expression Analysis with Transductive Parameter Transfer". *ACM Multimedia*, 2014 [7] *(*equal contribution)*
  - **G Zen**, E Sangineto, E Ricci and N Sebe. "Unsupervised Domain Adaptation for Personalized Facial Emotion Recognition". *International Conference on Multimodal Interaction (ICMI)*, 2014 [8]

– **G Zen***, L Porzi*, E Sangineto, E Ricci, N Sebe. "Transductive Parameter Transfer for Facial Expression and Gesture Recognition". *Submitted to IEEE Transaction on Multimedia (TMM)*, January 2015 *(*equal contribution)*

The following papers were published during the course of the Ph.D but not included in this thesis:

– **G Zen**, N Rostamzadeh, J Staiano, E Ricci, N Sebe. "Enhanced semantic descriptors for functional scene categorization". *IEEE International Conference on Pattern Recognition (ICPR)*, 2012 [9]

– N Rostamzadeh, **G Zen**, I Mironica, J Uijlings and N Sebe. "Daily living activities recognition via efficient high and low level cues combination and fisher kernel representation". *International Conference on Image Analysis and Processing (ICIAP)*, 2013 [10]

– A Gaidon, **G Zen**, J-A Rodriguez. "Self-Learning Camera: Autonomous Adaption of Object Detectors to Unlabeled Video Streams". *arXiv preprint arXiv:1406.4296*, 2014 [11]

Abbreviations

AE	**A**uto **E**ncoder
EMD	**E**arth **M**over's **D**istance
DDP-HMM	**D**ependent **D**irichlet **P**rocess **H**idden **M**arkov **M**odel
LP	**L**inear **P**rogram
MAS	**M**ultiscale **A**nomaly **S**core
MoG	**M**ixture **o**f **G**aussians
NMF	**N**on-Negative **M**atrix **F**actorization
PLSA	**P**robabilistic **L**atent **S**emantic Analysis
PTM	**P**robabilistic **T**opic **M**odel
TPT	**T**ransductive **P**arameter **T**ransfer

Chapter 1

Introduction

Interpreting a visual scene, including tasks such as localizing objects or recognizing actions, can be very easy for a person, but extremely challenging to be performed automatically by an artificial intelligence system. For example, if we look at Fig. 1.1 we can immediately tell where the ball is even if it is partially occluded. The reason why we can be sure of our anwser is because for solving this task we are not only deploying our sight, but we are also reasoning about the context and using our *a priori* knowledge. Indeed, for the scene to make sense, the ball has to be behind the player's head, on his right hand - which is also occluded. In other words, the key for developing an automatic vision system, does not simply lie in teaching to recognize objects or actions, but also in providing the capacity to reason about the events. In fact, it is not practically feasible to provide all the possible examples of how a ball looks like, as well as all the possible ways in which an action can be performed, but we can provide a system with the capacity to recognize an object or an action in an unseen configuration by leveraging on the context or on its *a priori* knowledge.

FIGURE 1.1: Image depicting a sample case for a task, *i.e.* "*can you tell where is the ball?*", which is extremly easy to perform for a human observer, but which can be very challenging for an artificial intelligence system.

In this work, we show that considering the correlation among atomic events in a scene (*i.e.* visual words) allows achieving higher performances in the task of discovering high level patterns of behavior within a learning framework based on the bag-of-words approach. In our novel method, the correlation among words is taken into account by deploying an efficient version of the Earth Mover's Distance (EMD). In this thesis, we present different variants of our method which better suit different case scenarios, and we discuss the advantages and disadvantage for each variant. For example, we discuss the concept of anomalous behavior w.r.t. the time granularity at which a scenario is observed, and we present a novel approach for long term analysis of a complex urban scenario, which is especially meant to deal with noise in the extraction of low level cues (i.e. foreground).

Unsupervised methods for the discovery of typical patterns of behavior are especially effective in scenarios such as the ones of video surveillance, in which the categories of behaviors and possible anomalies are unknown in advance (*i.e.* different traffic patterns) or sport games, where different possible players configuration have to be discovered. Conversely, when the categories of interest are known in advance, supervised approaches are usually preferred. Still, a very well known limitation of supervised approaches is due to the *dataset bias* effect. In [12], a significant drop in performance is observed when object detectors are trained on one generic dataset (*e.g.* PASCAL) and applied to another generic one (*e.g.* SUN09). In order to reduce the effort of collecting new label samples for adapting the classifiers to a new unseen domain, many approaches have been proposed. These approaches include both adapting the classifier to the new domain, as well as mapping the points from the target distribution to a new space in order to compensate the distribution mismatch between the *source* and *target* domains. In the case scenario of user gestures and facial analysis recognition, it has been shown that personalized models work better than generic ones. In this work, we consider each user as a specific domain, and we show that a personalized classifier can be learned by exploiting the similarity shared among users. Our contributions are described with more details in the Sec. 1.1.

1.1 Contribution of this work

The main contributions of our work are detailed as follows.

A Prototype Learning Framework for Discovering High Level Patterns of Behavior. We formulate the task of mining typical behaviors in dynamic scenes as a prototype learning problem. Our approach is based on a convex optimization problem, specifically a Linear Program (LP), thus, it is not prone to local minima and it is easy to implement. We show that similarity among visual words can be taken into consideration by using the EMD distance in the loss function.

To run experiments on medium-large scale datasets, following the idea in [13] we consider some efficient variations of EMD which use L_1 norm and its variants for ground distance definition. In this case, the flow network involved in the computation of the EMD is simplified and a the words' sorting has to be defined because only the similarity among adjacent bins is considered. To compute automatically the order of atomic activities, a novel strategy based on simulated annealing is proposed.

We show how salient patterns at multiple scales can be discovered. Differently from previous works and thanks to the theory of Parametric LP [14, 15], our algorithm performs a multiscale temporal segmentation of the video scene in one shot. Comparing salient aspects extracted at multiple scales can also be useful in individuating anomalous patterns. To this aim we propose a Multiscale Anomaly Score (MAS).

We evaluate extensively our approach on four datasets (three of which are publicly available), showing that it offers competitive performance w.r.t. state-of-the-art methods. Our code and the results from our experiments were made available to the community (http://disi.unitn.it/∼zen).

Discovering Typical Patterns of Behavior in Crowded Noisy Scenes. Inspired by similar motivations as in [1–3], we develop a novel approach for complex scene analysis which is specifically tailored to cope with the uncertainty and the noise arising in visual modeling of complex dynamic scenes. Differently from previous works [1, 3, 16–19], we model the task of extracting salient activities as a matrix factorization problem and we consider as objective function the Earth Mover's Distance (EMD) [20], which is well-known to be a robust metric in case of comparison on noisy histograms. To further reduce the influence of noisy data we also constrain the computed vector basis to be sparse. In surveillance videos, as the one we considered in this thesis (Ch. 2,3), where scenes have multiple temporal activity patterns happening simultaneously, a sparsification procedure is crucial for semantic scene interpretation purposes, helping to identify the atomic

activities which are distinctive of a specific high level behavior. To our knowledge, no previous works have addressed complex video scene analysis under a sparse coding framework.

We proposed a novel weakly supervised EMD-NMF framework for discovering typical patterns in dynamic scenes. However, while in [3, 4] the distance among visual words is defined a priori, we jointly learns the ground distance parameters and the high level patterns in a discriminative fashion. Our EMD-NMF algorithm is general and can be used in other applications not limited to the computer vision field, such as data mining and clustering.

Furthermore, we consider the case of long-term analysis of complex dyinamix scene. We show how images available on the web from surveillance webcams can be used as sensors in urban scenarios for monitoring and interpreting states of interest such as traffic intensity. We highlight the power of the cyclical aspect of the lives of people and of cities. We extract from long-term streams of images typical patterns of behavior and anomalous events and situations, based on considerations of day of the week and time of day. In order to deal with the high variability of background appearance, due to the challenging illumination and weather conditions, we propose a novel framework for background subtraction based on sparse coding which is especially tailored to cope with long term noisy sequence of webcam data.

Personalizing Models for Facial Expression Analysis with Transductive Parameter Transfer. We propose a novel domain adaptation approach to obtain personalized models for facial expression analysis. To the best of our knowledge, this is the first approach for *Transductive Parameter Transfer (TPT)*, where the parameters of the source classifiers are "transferred" to the target domain using a regression framework without the need of labeled target data. Previous methods either rely on *instance transfer* (source sample selection or re-weighting) or look for a *shared feature space* between sources and target data [21]. Our approach is is significantly faster (and simpler to implement) than other domain adaptation approaches, most of which are based on time consuming retraining strategies. We show that we can compute the target classifier in significantly shorter time, and we believe that this is an important aspect for user personalization in real HCI applications.

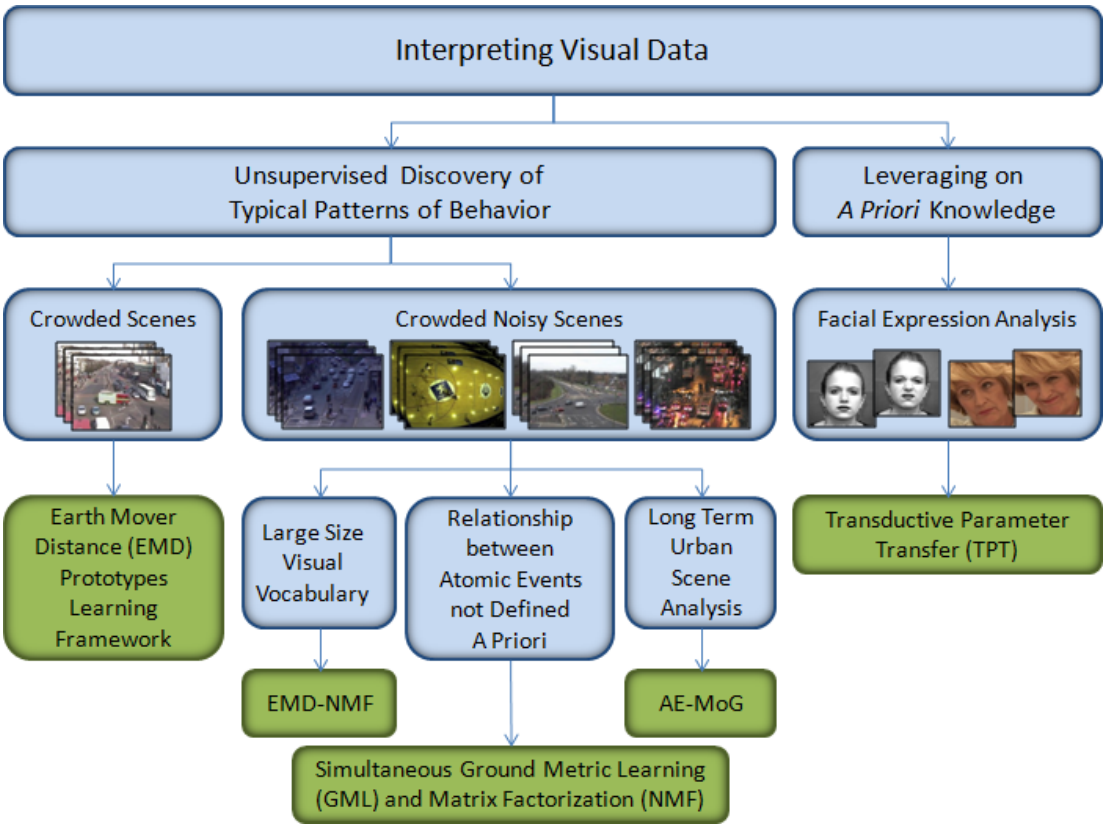


FIGURE 1.2: Overview of our work in the context of visual data interpretation. Frames highlighted in blue indicate different case scenarios considered. Our proposed approaches are highlighted in green.

We propose accurate approaches to quantify the difference between source and target distributions which are based on specific kernels for distributions. In particular, restricting the chosen classifiers to be linear SVMs, we propose to represent the *source* data distributions by means of the corresponding Support Vectors, which are related with $(\mathbf{w}_i, b_i)$ via the Karush-Kuhn-Tucker (KKT) condition. This makes stronger the geometric relation between $(\mathbf{w}_i, b_i)$ and our non-parametric representation of the source data.

The proposed domain adaptation approach is a general framework that can be applied to different types of data, e.g. images, audio, text, physiological signals, inertial measurements, etc. We show that by testing our method on a non-visual dataset, *i.e.* recognition of accelerometer based smartwatch gestures. Besides, while the proposed approach was originally meant to work with linear SVM, we show that the method can be extended to deal with semi-parametric non-linear classifiers. Specifically we consider a Distance Learning (DL) algorithm [22] and we call this extension TPTDL.

1.2 Outline

The remaining part of this thesis is organized as follows. In Chapters 2 and 3 we present our work about unsupervised discovery of typical patterns of behavior in complex scenes. In Chapter 4 we present our approach for learning personalized classifiers. Finally, conclusions are drawn in Chapter 5. A graphical overview of the proposed methods and considered scenarios is shown in Fig. 1.2.

Chapter 2

A Prototype Learning Framework for Discovering Typical Patterns of Behavior

2.1 Intuition

In the last decades, many efforts have been devoted to develop methods for automatic scene understanding in the context of video surveillance applications. We presents a novel non-object centric approach for complex scene analysis. Similarly to previous methods, we use low level cues to individuate atomic activities and create clip histograms. Differently from recent works, the task of discovering high-level activity patterns is formulated as a convex prototype learning problem. This problem results into a simple linear program that can be solved efficiently with standard solvers. The main advantage of our approach is that, using as objective function the Earth Mover's Distance (EMD), the similarity among elementary activities is taken into account in the learning phase. To improve scalability we also consider some variants of EMD adopting L_1 as ground distance for one and two dimensional, linear and circular histograms. In these cases only the similarity between neighboring atomic activities, corresponding to adjacent histogram bins, is taken into account. Therefore we also propose an automatic strategy for sorting atomic activities. Experimental results on publicly available

datasets show that our method compares favorably with state-of-the-art approaches, often outperforming them.

2.2 Related Work

The approaches for complex scenes analysis without object tracking/detection have recently gained an increasing popularity [17, 18, 23, 24]. Most of these methods adopt a probabilistic framework: a word-document paradigm is employed to represent the co-occurrences of atomic events and sophisticated PTMs are used to extract salient activities (topics). These approaches, specifically developed for unsupervised scene analysis, have several advantages over standard clustering techniques (*e.g. k-means*), such as a greater flexibility to model complex tasks and the ability to infer spatio-temporal dependencies among discovered activities. The approach we propose is significantly different from PTMs-based methods. In this work the task of discovering high-level activity patterns is formulated as a Parametric LP. This permits not only to avoid the typical local minima problems but, more interestingly, to efficiently compute, under special conditions, the so-called regularization path associated to the LP. This means that we can explore the most k relevant activities for all possible values of k at roughly the same time as for one fixed value $k = \hat{k}$. In other words a multiscale video scene analysis arises naturally using our approach.

A large number of works in video analysis adopts a bag-of-words representation, not only in the context of complex scene analysis [25, 26] but also for related tasks such as human action recognition [27, 28]. This representation, while being very powerful, ignores the spatio-temporal arrangement of elementary features. Differently our approach explicitely focuses on exploiting atomic activity dependencies.

The most similar work to ours in the context of video scene understanding is perhaps [26]. In [26] a multiscale analysis is also proposed and diffusion maps are used in a preprocessing step before clustering. Differently our multi-resolution analysis is obtained during the clustering phase and it is also used for individuating unusual behaviors. Being able to detect anomalous patterns is of fundamental importance not only in visual surveillance applications [29, 30], but in many other contexts (see [31] for a review). Our MAS is related to previous nonparametric outlier mining techniques where the global and

local density of the data are used to define the so-called outlier factors [32, 33]. However, MAS is novel since it is specifically tailored to the proposed clustering algorithm, aiming to quantify how the clusters size changes at subsequent scales. Previous approaches [29, 30] do not exploit multiscale segmentation levels for detecting unusual behaviours.

Our approach draws its inspiration from sparse signal approximation algorithms such as the fused lasso [34]. However, to the best of our knowledge we are the first to adopt a similar strategy for mining complex video scenes and to show that parametric LP can be a useful tool for multiscale analysis. To compute the entire solution path we resort on the approach described in [35]. However, our clustering algorithms are novel with respect to sparse signal approximation methods in [35]. In particular EMD has never been used in this context. This choice is motivated by the fact that with noisy histogram data the EMD is a better metric with respect to bin-to-bin distances.

Our work is related to [36] where EMD is used in the objective function of an optimization problem. However, in [36] the authors focused on Nonnegative Matrix Factorization. Finally recent clustering methods [37–39] are also closely related to our approach. In [37, 38] two algorithms for clustering with EMD are also presented, while in [39] the link between sensitivity analysis in LP and multiscale clustering is exploited. However, these works, not developed in the context of dynamic scene analysis, rely on optimization problems which are significantly different from ours.

2.3 Method

2.3.1 Discovering Spatio-Temporal Patterns in Dynamic Scenes

This Section gives an overview of the proposed approach for discovering high-level activity patterns in dynamic scenes.

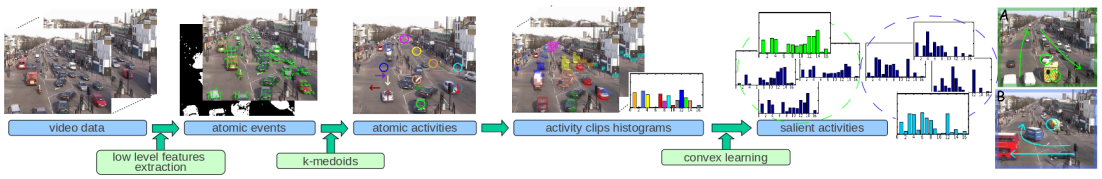


FIGURE 2.1: Flowchart of the proposed approach

In the first phase (Fig. 2.1) low level features for each pixel are extracted from the video, *i.e.* for the foreground/background information and the optical flow are computed. As background subtraction algorithm we use a simple dynamic Gaussian-Mixture background model [40], for the optical flow we used the Lucas-Kanade algorithm. By thresholding the magnitude of the flow vector foreground pixels are divided into static and moving pixels. For moving pixels we also quantize the optical flow into $n_\theta = 8$ directions. Then we divide the scene into $p \times q$ patches. For each patch we build a patch descriptor vector $\boldsymbol{v} = [x \ y \ f_g \ \bar{d}_{of} \ \bar{m}_{of}]$ where (x, y) denotes the coordinates of the patch center in the image plane, f_g is the percentage of foreground pixels in the patch, $\bar{d}_{of}$ is the mode of the optical flow orientations distribution and $\bar{m}_{of}$ is the average magnitude of optical flow vectors with direction $\bar{d}_{of}$. For patches of static pixels we set $\bar{d}_{of} = \bar{m}_{of} = 0$. To limit the influence of noise in low level features extraction we discard patches with few pixels of foreground, *i.e.* such that $f_g \leq T_{fg}$. We define an *atomic event* as a valid patch descriptor $\boldsymbol{v}$.

In the second phase a codebook of *atomic activities* is constructed. To this aim we define the following distance function between two atomic events $\boldsymbol{v}_q = [x^q \ y^q \ \bar{q}^q_{of} \ \bar{m}^q_{of}]$ and $\boldsymbol{v}_t = [x^t \ y^t \ \bar{d}^t_{of} \ \bar{m}^t_{of}]$ as

$$\delta_{qt} = \alpha \Delta p + (1 - \alpha)(\Delta m + \Delta \theta) \tag{2.1}$$

where:

$$\begin{aligned}
\Delta p &= \sqrt{(x^t - x^q)^2 + (y^t - y^q)^2} \\
\Delta m &= |\bar{m}^t_{of} - \bar{m}^q_{of}| \\
\Delta \theta &= \begin{cases} 0 & \text{if } \bar{m}^q_{of} = 0 \vee \bar{m}^t_{of} = 0 \\ \min(|\bar{d}^t_{of} - \bar{d}^q_{of}|, n_\theta - |\bar{d}^t_{of} - \bar{d}^q_{of}|) & \textit{otherwise} \end{cases}
\end{aligned}$$

In practice the parameter α in (2.1) controls the relative importance of position and motion information. In our experiments we set $\alpha = 0.5$. Then we group atomic events using K-medoids clustering. Each cluster represents an atomic activity. Subsequently we divide the video into short video clips and for each clip c we construct an *activity histogram* $\boldsymbol{h}_c$ representing the distribution of atomic activities. In the last phase the video clips are grouped according to their similarity. We propose a novel algorithm which, given a training set of clips histograms, outputs a small set of histograms constituting a

synthetic representation of the original data. These histogram prototypes represent the *salient activities* occurring in the scene.

2.3.2 Earth Mover's Prototypes

In this Section we present our Earth Mover's prototypes learning approach. Some basic concepts about EMD and its variations are given beforehand respectively in Sec.2.3.2.1 and 2.3.2.2.

2.3.2.1 Earth Mover's Distance

The Earth Mover's Distance (EMD) [20] $\mathcal{D}_E(\boldsymbol{h}, \boldsymbol{p})$ between two histograms $\boldsymbol{h}, \boldsymbol{p}$ normalized to unit mass is obtained as the solution of the following transportation problem:

$$\min_{f_{qt} \geq 0} \sum_{t,q=1}^{D} d_{qt} f_{qt} \quad \text{s.t.} \quad \sum_{q=1}^{D} f_{qt} = h^t, \ \sum_{t=1}^{D} f_{qt} = p^q \tag{2.2}$$

The variable f_{qt} denotes a flow representing the amount transported from the q-th supply to the t-th demand and d_{qt} the ground distance between q and t. Usually d_{qt} is defined by L_1 or L_2 distance. Figure 2.2(a) depicts the flow network associated to EMD. The problem (4.14) is a LP which can be solved efficiently due to the special structure of its sparse constraints [13, 20]. However, in the case of high dimensional histograms solving (4.14) can be very time consuming due to the large number of flow variables involved.

2.3.2.2 Linear, Circular and Thresholded EMD-L_1

Several methods have been proposed in the past to speed up the EMD distance computation. In [13], it is observed that, for histograms normalized to unit mass and L_1 ground distance (*i.e.* $d_{qt} = |q - t|$), every positive flow between faraway histograms bins can be replaced by a sequence of flows between neighbor bins. This implies that, for *unidimensional histograms* (*i.e.* $\boldsymbol{h}, \boldsymbol{p} \in \mathbb{R}^D$), formula (4.14) can be simplified:

$$\begin{aligned} \min \quad & \sum_{q=1}^{D-1} f_{q,q+1} + \sum_{q=2}^{D} f_{q,q-1} \\ \text{s.t.} \quad & f_{q,q+1} - f_{q+1,q} + f_{q,q-1} - f_{q-1,q} = b^q \ , \quad b^q = h^q - p^q \qquad \forall q, q = 1 \dots D \\ & f_{q,q+1}, f_{q,q-1} \geq 0 \end{aligned} \tag{2.3}$$

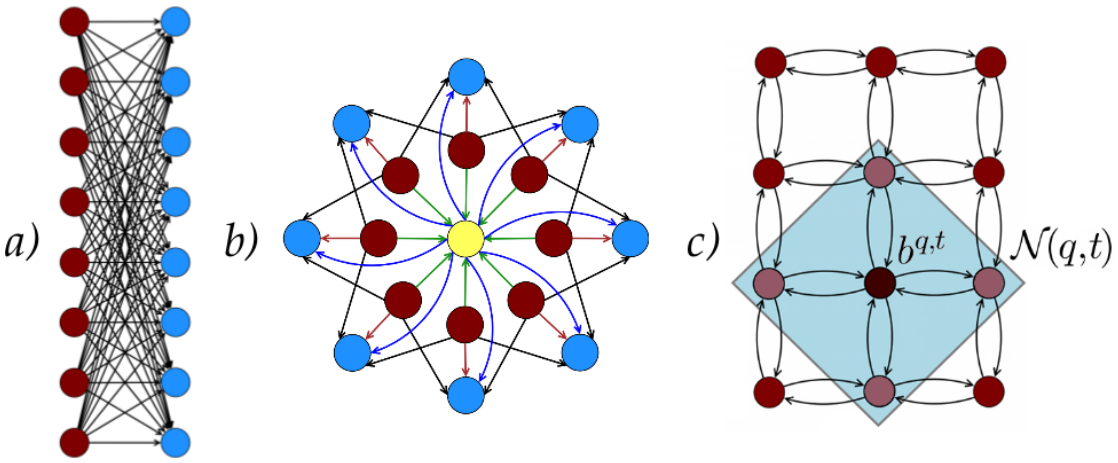


FIGURE 2.2: The flow networks associated to **(a)** EMD, **(b)** EMD with L_1 ground distance, **(c)** EMD with thresholded L_1 ground distance for circular histograms. In **(b)** the yellow node is the transhipment vertex. Ingoing edge (green) cost is the threshold (2 in this case) and outgoing edge (blue) cost is 0. Red edges have cost 0. Black edges are 1-cost edges. **(d)** The two dimensional grid associated to (2.4).

The number of flow variables reduces from $O(D^2)$ in (4.14) to $O(D)$. This is greatly beneficial in terms of computational cost since the number of variables is a dominant factor in the time complexity of all LP algorithms. Moreover, the number of equality constraints is reduced by half and all the ground distances involved in the EMD-L_1 are ones. This is practically useful saving multiplications during computation. Eqn. (2.3) considers unidimensional histograms but the EMD-L_1 can be defined also for higher dimensional cases [41]. For example for *two-dimensional histograms* (*i.e.* $\boldsymbol{h}, \boldsymbol{p} \in I\!R^{D_1 \times D_2}, D_1 D_2 = D$) the only difference is that the neighborhood structure is not a line but a grid. The resulting optimization problem is:

$$\begin{aligned} &\min_{f_{m,n;q,t} \geq 0} \sum_{q,t} \sum_{m,n \in \mathcal{N}(q,t)} f_{q,t;m,n} \\ &\text{s.t.} \quad \sum_{m,n \in \mathcal{N}(q,t)} f_{q,t;m,n} - \sum_{m,n \in \mathcal{N}(q,t)} f_{m,n;q,t} = b^{q,t} \ \forall q,t \end{aligned} \tag{2.4}$$

where $b^{q,t} = h^{q,t} - p^{q,t}$, the indices q, t correspond to the position of a bin while its neighborhood $\mathcal{N}(q,t)$ is represented by the four adjacent bins (see Fig.2.2.c). In [42, 43] other computationally efficient variations of EMD have been proposed. In [43] the EMD with thresholded L_1 ground distance (*i.e.* $d_{qt} = \min(|q-t|, 2)$) is considered for robust comparison of noisy histograms. The adoption of the threshold implies the introduction of a transhipment vertex, slightly increasing the number of flow variables [43]. However, it has been shown that saturated distances are beneficial in terms of accuracy results in several applications. In [42] the same authors proposed a *circular histogram* representation. In this case a different ground distance is needed, *i.e.* $d_{qt} =$

$\min(\min(|q-t|, D-|q-t|), 2)$. With thresholded ground distance and circular histograms, (4.14) assumes the form:

$$\begin{aligned} \min \quad & \sum_{q=1}^{D} f_{q,q+1} + \sum_{q=1}^{D} f_{q,q-1} + 2\sum_{q=1}^{D} f_{q,D+1} \\ \text{s.t.} \quad & f_{q,q+1} - f_{q+1,q} + f_{q,q-1} - f_{q-1,q} + f_{q,D+1} = h^q - p^q \\ & f_{q,q+1}, f_{q,q-1}, f_{q,D+1} \geq 0 \end{aligned} \tag{2.5}$$

where the flow variables $f_{q,D+1}$ correspond to the links connecting sources to the transhipment vertex. Figure 2.2(b) depicts the associated flow network. In practice with respect to (2.3) in (2.5) also flows between sources and the transhipment vertex are considered. However, the number of flow variables is still $O(D)$.

2.3.2.3 Convex Optimization for Prototypes Learning

Given a set of histograms $\mathcal{H} = \{\boldsymbol{h}_1, \ldots, \boldsymbol{h}_N\}$, the task of prototype learning is the problem of computing a set $\mathcal{P} = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_N\}$, such that the following two requirements are jointly satisfied:

- each prototype $\boldsymbol{p}_i$ must be as much similar as possible to the associated histogram $\boldsymbol{h}_i$
- the set of prototypes is a sparse representation of the original dataset $\mathcal{H}$ (*i.e.* the number of different prototypes must be small)

The prototype learning problem can be formalized as follows:

$$\begin{aligned} \min \quad & \sum_{i=1}^{N} \mathcal{L}(\boldsymbol{h}_i, \boldsymbol{p}_i) + \lambda \sum_{i \neq j,\ i,j=1}^{N} \eta_{ij} \max_{q=1\ldots D} |p_i^q - p_j^q| \\ \text{s.t.} \quad & \boldsymbol{p}_i \geq 0, \qquad \sum_t p_i^t = 1 \quad \forall i = 1 \ldots N \end{aligned} \tag{2.6}$$

where the constraints ensure that the computed prototypes are histograms normalized to unit mass. The objective function consists of two terms. The loss function $\mathcal{L}(\cdot)$ penalizes the difference between the original histograms and the associated prototypes. In this work we focus on the specific form of (2.6) in which $\mathcal{L}(\cdot)$ is a convex function. The second term is meant to minimize the number of different prototypes. In fact the adoption of

the $L_1 - L_\infty$ norm induces sparsity, thus producing a small number of prototypes. The set of binary coefficients $\eta_{ij} \in \{0, 1\}$ indicates the pairs of histograms which must be merged. In the absence of prior knowledge, for each histogram $\boldsymbol{h}_i$ a set of N_P nearest neighbors can be identified and the associated η_{ij} set to 1 if $\boldsymbol{h}_j$ is a neighbor of $\boldsymbol{h}_i$. In alternative, temporal dependencies can be encoded into η_{ij}: for example if histograms represent temporally adjacent clips it is reasonable to set $\eta_{ij} = 1$ if $i = j-1, j = 2 \dots N$, $\eta_{ij} = 0$ otherwise. The relative importance of loss and regularization is controlled by the positive coefficient λ. When $\lambda = 0$ all prototypes $\boldsymbol{p}_i$ must be equal to their corresponding histograms $\boldsymbol{h}_i$ while for $\lambda \to \infty$ all prototypes should be equal to each others. For $0 \leq \lambda < \infty$ a number of prototypes k between N and 1 can be obtained. In truth, for large values of λ and few prototypes the L_1 norm also induces the prototypes to be quite similar to each other. In practice as λ decreases the effect of the loss function is stronger and the computed prototypes are quite different.

2.3.2.4 Learning Prototypes with EMD

In this work we present a specific formulation of (2.6) where the EMD is adopted as loss function:

$$\begin{aligned} \min \quad & \sum_{i=1}^{N} \mathcal{D}_E(\boldsymbol{h}_i, \boldsymbol{p}_i) + \lambda \sum_{i \neq j} \eta_{ij} \max_{q=1\dots D} |p_i^q - p_j^q| \\ \text{s.t.} \quad & \boldsymbol{p}_i \geq 0, \qquad \sum_t p_i^t = 1 \quad \forall i = 1 \dots N \end{aligned} \tag{2.7}$$

Therefore to compute the prototypes we introduce the EMD formulation (4.14) into (2.7) and we get the following LP:

$$\begin{aligned} \min_{p_i^q, f_{qt}^i, \zeta_{ij} \geq 0} \quad & \sum_{i=1}^{N} \sum_{t,q=1}^{D} d_{qt} f_{qt}^i + \lambda \sum_{i \neq j} \eta_{ij} \zeta_{ij} \\ \text{s.t.} \quad & -\zeta_{ij} \leq p_i^q - p_j^q \leq \zeta_{ij}, \ \forall q, \forall i, j \ i \neq j \\ & \sum_{q=1}^{D} f_{qt}^i = h_i^t, \ \forall t \quad \sum_{t=1}^{D} f_{qt}^i = p_i^q, \ \forall q, \forall i \end{aligned} \tag{2.8}$$

Note that the constraints $\sum_t p_i^t = 1$ are removed since they are automatically satisfied as the original histograms are normalized. It is worth noting that at the coordinate level we adopt the L_∞ norm rather than the L_1 norm. This does not promote sparsity

but produces the effects that all coordinates of a prototype go to zero together and significantly reduces the computational cost of solving (2.8) limiting the number of slack variables ζ_{ij}.

Regarding the ground distance d_{qt} definition, we use the fact that each histogram bin corresponds to an atomic activity q, which is represented by the associated centroid $\boldsymbol{c}_q = [x^q \; y^q \; \bar{q}^q_{of} \; \bar{m}^q_{of}]$ computed by K-medoids in the first phase of our approach. Therefore we define the ground distance between two atomic activities $\boldsymbol{c}_q = [x^q \; y^q \; \bar{d}^q_{of} \; \bar{m}^q_{of}]$ and $\boldsymbol{c}_t = [x^t \; y^t \; \bar{d}^t_{of} \; \bar{m}^t_{of}]$ as follows:

$$d_{qt} = \alpha\Delta p + \beta(\Delta m + \Delta\theta) + (1 - \alpha - \beta)(1 - \Delta T_C) \tag{2.9}$$

where the terms $\Delta p, \Delta m$ and $\Delta\theta$ are defined as in (2.1). The last term ΔT_C takes into account the temporal correlation between atomic activities: starting from a training set of activity histograms $\{h_1, \ldots, h_{N_c}\}$, where N_c is a fixed number of clips, we consider, for each pair $\boldsymbol{c}_q, \boldsymbol{c}_t$ of atomic activities, the vectors $H_q = (h^q_1, \ldots, h^q_{N_c})$ and $H_t = (h^t_1, \ldots, h^t_{N_c})$ and set ΔT_C equal to the correlation coefficient between H_q and H_t. In (2.9) the ground distance depends on two parameters, α and β which control the relative importance of position, motion and temporal correlation.

2.3.2.5 Speeding up Prototype Learning

For large N and D solving (2.8) is still time consuming even for today's sophisticated LP solvers. The computational cost is especially high due to the large number of flow variables f^i_{qt}. Actually we do not specifically need them since we are only interested in computing the prototypes $\boldsymbol{p}_i$. Therefore to speed up calculations we also propose to modify (2.8) as follows.

We consider the special case of EMD with L_1 distance over bins as ground distance. In our specific application the idea is that similar atomic activities should correspond to neighboring bins in activity histograms. To this aim, the atomic activities are sorted according to the associated location and motion information (see subsection 2.3.3). With this premises, we propose to simplify (2.8) using the efficient formulation of EMD (2.3).

So substituting the definition of EMD-L_1 (2.3) into (2.7) we get:

$$\begin{aligned} \min \quad & \sum_{i=1}^{N}\left(\sum_{q=1}^{D-1} f_{q,q+1}^{i} + \sum_{q=2}^{D} f_{q,q-1}^{i}\right) + \lambda \sum_{i\neq j} \eta_{ij}\zeta_{ij} \\ \text{s.t.} \quad & -\zeta_{ij} \leq p_i^q - p_j^q \leq \zeta_{ij},\ \forall q,\ \forall i,j,\ i \neq j \\ & f_{q,q+1}^{i} - f_{q+1,q}^{i} + f_{q,q-1}^{i} - f_{q-1,q}^{i} = h_i^q - p_i^q,\ \forall q,\ \forall i \\ & p_i^q, f_{q,q+1}^{i}, f_{q,q-1}^{i}, \zeta_{ij} \geq 0 \end{aligned} \tag{2.10}$$

The resulting optimization problem is a LP with $n_{var} = n_f + n_p + n_\zeta = 2N(D-1) + ND + \frac{1}{2}NN_P$ variables, in case we adopt the nearest neighbor approach for setting the coefficients $\eta_{ij} = 1$. In this case for large datasets and small histograms ($N \gg D$) the computational cost of (2.10) is dominated by the number of slack variables. However, by considering a small number of neighbors N_P, (2.10) can be solved efficiently even for large datasets. Analogously a prototype learning approach can be devised for two dimensional histograms by considering the EMD-L_1 definition (2.4).
Similarly, for circular histograms and EMD with thresholded L_1 ground distance, the prototype learning algorithm can be obtained by inserting (2.5) in (2.7):

$$\begin{aligned} \min \quad & \sum_{i,q} f_{q,q+1}^{i} + \sum_{i,q} f_{q,q-1}^{i} + 2\sum_{i,q} f_{q,D+1}^{i} + \lambda \sum_{i\neq j} \eta_{ij}\zeta_{ij} \\ \text{s.t.} \quad & -\zeta_{ij} \leq p_i^q - p_j^q \leq \zeta_{ij},\ \forall q,\ \forall i,j,\ i \neq j \\ & f_{q,q+1}^{i} - f_{q+1,q}^{i} + f_{q,q-1}^{i} - f_{q-1,q}^{i} + f_{q,D+1}^{i} = h_i^q - p_i^q \\ & p_i^q, f_{q,q+1}^{i}, f_{q,q-1}^{i}, f_{q,D+1}^{i}, \zeta_{ij} \geq 0 \end{aligned} \tag{2.11}$$

The resulting optimization problem is a LP with $n_{var} = 4ND + \frac{1}{2}NN_P$.

2.3.2.6 Learning Prototypes with bin-to-bin Distances

To demonstrate the advantages of considering cross-bin similarities when learning prototypes, we briefly discuss the form that (2.6) assumes when bin-to-bin distances are used as metrics and some related approaches in the literature. For example when the L_1 norm is chosen as loss function, (2.6) assumes the form:

$$\begin{aligned} \min \quad & \sum_{i=1}^{N}\sum_{q=1}^{D} |h_i^q - p_i^q| + \lambda \sum_{i\neq j} \eta_{ij} \max_{q=1\ldots D} |p_i^q - p_j^q| \\ \text{s.t.} \quad & \boldsymbol{p}_i \geq 0, \qquad \sum_t p_i^t = 1 \quad \forall i = 1\ldots N \end{aligned} \tag{2.12}$$

The resulting optimization is still a LP (as in the case of EMD) and can be solved efficiently with standard solvers once slack variables have been introduced. The proposed approach (2.6) can also be used with Kullback-Leibler distance as loss functions:

$$\min \quad \sum_{i=1}^{N}\sum_{q=1}^{D} h_i^q \log \frac{h_i^q}{p_i^q} + \lambda \sum_{i \neq j} \eta_{ij} \max_{q=1\ldots D} |p_i^q - p_j^q|$$

Similarly to L_1 and KL, also the L_2 norm can be used in the loss function in (2.6). In particular, if a sum of L_1 norms rather than a combination of L_1-L_∞ is used as regularization term and no constraints are imposed on the prototypes $\boldsymbol{p}_i$, the following optimization problem is obtained:

$$\min \quad \sum_{i=1}^{N} ||\boldsymbol{h}_i - \boldsymbol{p}_i||^2 + \lambda \sum_{i \neq j} \eta_{ij} \sum_{q} |p_i^q - p_j^q| \tag{2.13}$$

The special case where $\eta_{ij} = 1$ if $i = j - 1$ and $\eta_{ij} = 0$ otherwise leads to the well known "total variation denoising" procedure [44] or to a special case of the fused lasso [34]. However, it is worth nothing that in our case the choice of using a L_1-L_∞ norm rather than a sum of L_1 is motivated by computational efficiency reasons. In fact since our optimization problem is an LP and we solve it with standard solvers, the number of slack variables is kept limited. In all these cases, only bin-to-bin comparisons are allowed. Indeed the experimental results presented in the Section 2.4 demonstrate that bin-to-bin distances are less effective than EMD when learning prototypes for dynamic scene understanding.

2.3.3 Ordering Atomic Activities

Elementary activities are not independent and it is desirable to take into account their similarity when learning activity prototypes. A straightforward way to impose this is to encode atomic activities similarity in the ground distance definition (2.9). This means considering similarity among all possible pairs of atomic activities and a high computational cost of solving (2.8) even for problems with a small N. A similar requirement can be imposed also in the case of the more efficient EMD variants based on L_1. In this case considering atomic activities similarity means sorting them according to a prespecified criterion. The idea is that, when constructing clip histograms, neighboring activities correspond to similar ones.

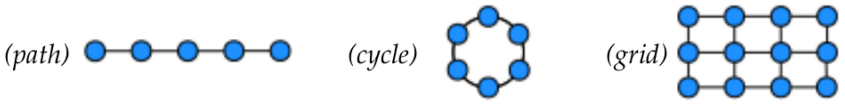


FIGURE 2.3: Structures used to arrange atomic activities.

Algorithm 1 Sorting atomic activities

Input: atomic activities $\boldsymbol{C} = \{\boldsymbol{c}_1, \dots \boldsymbol{c}_D\}$, graph $G(V, E)$, with $V = \{v_1, \dots, v_D\}$
$T \leftarrow T_0$ *initialize temperature*
$\sigma(\boldsymbol{c}_i) \leftarrow v_i, i = 1 \dots D$ *initialize* $\sigma()$
$\mathcal{D}_0 \leftarrow$ initial distortion *equation (2.14)*
$M \leftarrow 1$ *counter of accepted moves*
while $M > 0$
  $M \leftarrow 0$ *reset the counter*
  repeat N_{iter} times *generate move hypothesis*
    $\boldsymbol{c}_k \leftarrow$ randomly selected atomic activity
    $v_j \leftarrow$ randomly selected node from $V \setminus \{\sigma(\boldsymbol{c}_k)\}$
    $\boldsymbol{c}_i \leftarrow \sigma^{-1}(v_j)$
    $\sigma(\boldsymbol{c}_i) \leftarrow \sigma(\boldsymbol{c}_k)$
    $\sigma(\boldsymbol{c}_k) \leftarrow v_j$
    $\mathcal{D}_n \leftarrow$ compute distortion
    $\Delta\mathcal{D} \leftarrow \mathcal{D}_n - \mathcal{D}_0$;
    Accept move with probability $\min(e^{-\Delta\mathcal{D}/T}, 1)$
    if move accepted **then**
      $M \leftarrow M + 1$; $\mathcal{D}_0 \leftarrow \mathcal{D}_n$
  $T \leftarrow \eta * T$ *decrease the temperature* ($\eta < 1$)
end
Output: function $\sigma()$

To this aim we propose to find the best arrangement of the atomic activities into appropriate graph structures in order to minimize the distortion between the ground distances d_{qt} and the distances $\mathcal{D}_g$ of the nodes q and t within the graph (*i.e.* the length of the shortest path connecting them). As discussed at the beginning of this section, in this work we consider the three following graph structures: *path* graph, *cycle* graph and *square grid* graph (corresponding respectively to 1D, circular and 2D histograms, see Fig. 2.3), where the number of nodes is equal to the number of atomic activities. The distortion is defined as follows:

$$\sum_{q=1}^{D} \sum_{t=q+1}^{D} (d_{qt} - \mathcal{D}_g(\sigma(\boldsymbol{c}_q) - \sigma(\boldsymbol{c}_t)))^2 \tag{2.14}$$

which has to be minimized with respect to $\sigma()$, a one-to-one function mapping atomic activities to nodes of the graph. The minimization is achieved by Algorithm 1 which implements a simulated annealing approach. The temperature T_0 is set to a value such that a given fraction (about 0.75) of the moves would be initially accepted. The values of N_{iter} and η used in the experiments are 10000 and 0.99, respectively.

2.3.4 Multiscale Anomaly Score

A crucial property of (2.6) is that the sparsity achieved is controlled by a single parameter, *i.e.* the regularization constant λ. In other words, for λ varying between ∞ and 0, a different number of prototypes between 1 and N can be obtained. In the case of automatic scene understanding, this corresponds to discover different salient activities at multiple scales. For example, for traffic scene analysis, for large values of λ we can obtain a very rough description of the activities differentiating among clips with moving vehicles or clips corresponding to vehicles stopped at the traffic lights. As λ decreases we gradually enhance the level of details of the analysis differentiating among vehicles flows of different intensity.

Instead of finding the value of λ which provides the optimal prototypes we propose to exploit the solutions of (2.6) for different values of λ. More formally, given a set of N histograms $\boldsymbol{h}_i$ we first introduce the following characterization of sets of fused histograms as they are generated by our algorithms.

Definition 1. (Sets of Fused Histograms) *Let $\lambda = \bar{\lambda}$ and $\mathcal{H}_\ell^{\bar{\lambda}}$ be a set of histograms with $\ell = 1, \ldots, N(\bar{\lambda})$ where $N(\bar{\lambda})$ is the number of different prototypes obtained for $\lambda = \bar{\lambda}$. Then a valid set of fused histograms $\mathcal{H}_\ell^{\bar{\lambda}}$ satisfies the following properties:*

- $\cup_{\ell=1}^{N(\bar{\lambda})} \mathcal{H}_\ell^{\bar{\lambda}} = \mathcal{H}$
- $\mathcal{H}_\ell^{\bar{\lambda}} \cap \mathcal{H}_m^{\bar{\lambda}} = \emptyset$, $\forall \ell \neq m$.
- $\forall\ \boldsymbol{h}_\ell, \boldsymbol{h}_m \in\ \mathcal{H}_k^{\bar{\lambda}}$ *we have* $p_\ell^q = p_m^q\ \forall\ q = 1 \ldots D$
- $\forall\ \boldsymbol{h}_\ell\ \in\ \mathcal{H}_\ell^{\bar{\lambda}}$ *and* $\boldsymbol{h}_m\ \in\ \mathcal{H}_m^{\bar{\lambda}}\ \exists q:\quad p_\ell^q \neq p_m^q$

In a nutshell a set of fused histograms corresponds to histograms associated to the same prototype. Different sets of histograms are generated for different values of λ. Comparing clustering results at multiple scales (*i.e.* comparing sets of fused histograms for different values of λ) we can detect unusual behaviors corresponding to atypical histograms. To this aim we define for each $\boldsymbol{h}_\ell$ an associated anomaly score. The general idea behind this score is to monitor how the clusters size changes for decreasing values of λ. From $\lambda = \infty$ (where all the histograms are represented by a single prototype) to $\lambda = 0$ where each histogram corresponds to a different prototype, the anomaly score of

$\boldsymbol{h}_k$ can be computed as the sum of the ratios of the size of the clusters containing $\boldsymbol{h}_\ell$ at two subsequent scales. Analyzing multiple levels we can distinguish between cases where a cluster with a single histogram is merged at higher level with a small cluster and situations where it belongs to a big cluster: in the first case its anomaly score is higher. Formally:

Definition 2. (MAS) *Let* $\boldsymbol{h}_\ell \in \mathcal{H}_\ell^{\lambda_i}$ *and* $\boldsymbol{h}_\ell \in \mathcal{H}_{\ell'}^{\lambda_{i-1}}$ *with* $\lambda_{i-1} > \lambda_i$. *We define the* **Multiscale Anomaly Score (MAS)** *of the histogram* $\boldsymbol{h}_\ell$ *as:*

$$MAS = 1 - \frac{1}{NL} \sum_{i=2}^{L} \frac{|\mathcal{H}_{\ell'}^{\lambda_{i-1}}|}{|\mathcal{H}_\ell^{\lambda_i}|}$$

In practice, the most anomalous clips tend to get a higher MAS. Let us consider the case of a cluster made by a single clip. In this case the ratio in the MAS definition is very low (actually zero) until the clip is merged into a large cluster. The later it is merged, the smaller the ratio value is, thus the higher the MAS is.

Note that while large values of L may lead to more accurate estimates of MAS, this also increases the computational cost since (2.8), (2.10) and (2.11) must be solved L times. However, in the following we show how in the special case of temporal segmentation a multiscale analysis can be obtained with computational cost comparable with that of solving (2.8), (2.10) or (2.11) for a single value of λ. As a final remark, we should say that we experimentally observed that if two histograms are fused for a certain value of $\lambda = \bar{\lambda}$ (*i.e.* they belong to the same fused set) they will not necessarily remain fused for any $\lambda \geq \bar{\lambda}$. However, we found that for moderately large values of L, this does not decrease the accuracy of MAS analysis.

2.3.5 Multiscale Analysis in One Shot

In this Section we focus our attention on linear histograms and on the temporal segmentation approach *i.e.* we consider $\eta_{ij} = 1$ for $i = j - 1, j = 2 \dots N$ and $\eta_{ij} = 0$ otherwise. In particular we consider (2.8) and (2.12). We show that since (2.8) and (2.12) are parametric LP, an algorithm based on a variant of the revised simplex method can be developed to compute all possible sets of histogram prototypes for increasing values of λ.

2.3.5.1 Preliminaries: LP and Parametric LP

Given a matrix $\mathbf{A} \in I\!R^{n\times m}$ and the vectors $\mathbf{c} \in I\!R^m$, $\mathbf{b} \in I\!R^n$ a LP in standard form [14, 15] is given by:

$$\min_{\mathbf{x}\geq 0} \quad \mathbf{c}'\mathbf{x} \qquad \text{s.t.} \quad \mathbf{A}\mathbf{x} = \mathbf{b} \tag{2.15}$$

If the matrix $\mathbf{A}$ is of full rank n and the polyhedron $\mathcal{P} = \{\mathbf{x} : \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq 0\}$ is bounded and non-empty, the LP has a bounded optimal solution. Let $\mathcal{B} \in \mathcal{I} = \{1, \ldots, m\}$ be an ordered set of n column indexes. Let $\mathbf{A}_\mathcal{B}$ be the $n \times n$ sub-matrix of $\mathbf{A}$ whose i-th column is $\mathbf{A}_i$. The set $\mathcal{B}$ is called a *feasible basis* if $\mathbf{A}_\mathcal{B}$ is of full-rank and $\mathbf{A}_\mathcal{B}^{-1}\mathbf{b} \geq 0$. Since $\mathbf{A}$ is of full rank and the linear program is feasible, a feasible basis always exists. A column $\mathbf{A}_i$ with $i \in \mathcal{B}$ is called a basic column, otherwise it is called a non-basic column and belongs to the set $\mathcal{N} = \mathcal{I} - \mathcal{B}$. A *basic feasible solution* (bfs) $\mathbf{x}_\mathcal{B}$ of the LP corresponding to a feasible basis $\mathcal{B}$ is obtained by $\mathbf{x}_\mathcal{B} = \mathbf{A}_\mathcal{B}^{-1}\mathbf{b}$ and $\mathbf{x}_\mathcal{N} = \mathbf{0}$. A bfs is *optimal* if it corresponds to a solution of the LP. There is a bijection between bfs and vertices of $\mathcal{P}$. The *simplex* method systematically explores the extreme points (bfs) of $\mathcal{P}$, *i.e.* starting from an initial extreme point, until an optimal extreme point is found.

A parametric LP problem has the form:

$$\min_{\mathbf{x}\geq 0} \quad (\mathbf{c} + \lambda\mathbf{a})'\mathbf{x} \qquad \text{s.t. } \mathbf{A}\mathbf{x} = \mathbf{b} \tag{2.16}$$

with $\mathbf{a} \in I\!R^m$ and $\lambda \in I\!R$. In [35] Yao and Lee showed that many algorithms in machine learning and specifically the family of regularization problems with piecewise linear loss and L_1 penalties (such as L_1 SVM) can be written in the form of (2.16) and a variant of the simplex method can be used for solving (2.16) for all possible values of λ simultaneously.

2.3.5.2 Multiscale Analysis

Let $\boldsymbol{p}, \boldsymbol{\delta}_+, \boldsymbol{\delta}_- \in I\!R^{ND}$, $\boldsymbol{\zeta} \in I\!R^{N-1}$ and $\mathbf{H} \in I\!R^{ND}$ be the vector obtained concatenating the histograms in the training set (*i.e.* $\mathbf{H}' = (\boldsymbol{h}'_1, \ldots, \boldsymbol{h}'_N)$). We first define the following matrices: the block diagonal matrix $\mathbf{D} \in I\!R^{(N-1)D\times(N-1)}$, $\mathbf{D} = \text{diag}(\mathbf{-1})$ and $\mathbf{-1} \in I\!R^D$

and the block Toeplitz matrix $\mathbf{\Sigma} \in I\!R^{(N-1)D \times ND}$,

$$\mathbf{\Sigma} = \begin{pmatrix} \mathbf{I} & \mathbf{-I} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{-I} & \dots & \mathbf{0} \\ \vdots & \ddots & & & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{I} & \mathbf{-I} \end{pmatrix},$$

with $\mathbf{I}$, $\mathbf{0}$ and $-\mathbf{I} \in I\!R^{D \times D}$.

Proposition 1. *The following elements:*

$$\mathbf{x} = \begin{pmatrix} \boldsymbol{f}' & \boldsymbol{\zeta}' & \boldsymbol{p}' & \boldsymbol{\delta}_+{}' & \boldsymbol{\delta}_-{}' \end{pmatrix}$$
$$\mathbf{a}' = \begin{pmatrix} \boldsymbol{\omega}' & \mathbf{0}' & \mathbf{0}' & \mathbf{0}' & \mathbf{0}' \end{pmatrix}$$
$$\mathbf{c}' = \begin{pmatrix} \mathbf{0}' & \mathbf{1}' & \mathbf{0}' & \mathbf{0}' & \mathbf{0}' \end{pmatrix}$$
$$\mathbf{A} = \begin{pmatrix} \mathbf{0} & \mathbf{D} & \mathbf{\Sigma} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{D} & -\mathbf{\Sigma} & \mathbf{0} & \mathbf{I} \\ \mathbf{F} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{G} & \mathbf{0} & \mathbf{-I} & \mathbf{0} & \mathbf{0} \end{pmatrix} \quad \mathbf{b} = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{H} \\ \mathbf{0} \end{pmatrix}$$

with $\boldsymbol{f} \in I\!R^{ND^2}$, $\boldsymbol{\omega} \in I\!R^{ND^2}$, $\boldsymbol{\omega} = (\mathbf{d} \dots \mathbf{d})$, $\mathbf{d} \in I\!R^{D^2}$, $\mathbf{d} = (d_{11}, \dots d_{1D} d_{21} \dots d_{DD})$, $\mathbf{F}, \mathbf{G} \in I\!R^{ND \times ND^2}$ being two block diagonal matrices, $\mathbf{F} = \text{diag}(\mathbf{Q}), \mathbf{G} = \text{diag}(\mathbf{T}), \mathbf{Q}, \mathbf{T} \in I\!R^{D \times D^2}$, $\mathbf{Q} = \text{diag}(\mathbf{1}')$, $\mathbf{1}' \in I\!R^D$

$$\mathbf{T} = \begin{pmatrix} \mathbf{e'}_1 & \mathbf{e'}_1 & \dots & \mathbf{e'}_1 \\ \mathbf{e'}_2 & \mathbf{e'}_2 & \dots & \mathbf{e'}_2 \\ \vdots & \ddots & & \vdots \\ \mathbf{e'}_D & \mathbf{e'}_D & \dots & \mathbf{e'}_D \end{pmatrix}$$

with $\mathbf{e'}_i \in I\!R^D$ *is a vector of all 0 and 1 in the i-th position, define (2.8) in the standard form (2.16) of a parametric LP.*

Given a parametric LP problem in standard form all possible solutions $\bar{\mathbf{x}}$ for different values of λ can be computed. For this purpose in this work we use a variation of the algorithm proposed in [35] by considering a different variant of the simplex methods rather than the tableau simplex *i.e.* the revised simplex method with the lexico-min rule since it offers computational advantages for sparse LPs and avoid situations of

degeneracy. According to this, the basic column to exit the current basis $\mathcal{B}$ is selected according to the lexico-min rule: the column which exits the basis is $\mathbf{A}_\ell$, where ℓ is the index of the lexicographically smallest row $\mathbf{A}^i/u_i$, $u_i > 0$, $\mathbf{u} = \mathbf{A}_{\mathcal{B}}^{-1}\mathbf{A}_j$ and $\mathbf{A}^i$ denotes the i-th row of $\mathbf{A}_{\mathcal{B}}$. The index ℓ always exists, since otherwise $u_i \leq 0$ for all i and the problem is unbounded. The resulting algorithm is presented in Algorithm 2.

The main difference and the main issue when running Algorithm 2 is how to individuate an optimal bfs $\mathcal{B}_0$. This can be obtained using any feasible basic index set $\bar{\mathcal{B}}_0$ and running the standard simplex algorithm for the associated LP problem *i.e.* for $\mathbf{a} = \mathbf{0}$. The following proposition shows how a basic feasible set $\bar{\mathcal{B}}_0$ can be individuated for the proposed problem (2.8).

Proposition 2. *The set of indices* $\bar{\mathcal{B}}_0 = \mathcal{I}_1 \cup \mathcal{I}_2$ *with* $\mathcal{I}_1 = \{kD + 1 : k = 0, \ldots, ND - 1/D\}$, $\mathcal{I}_2 = \{ND^2 + N + k : k = 0, \ldots, 3ND - 1\}$ *individuates a bfs for (2.8).*

Proof. See Appendix A.

Algorithm 2 can generally be applied not only to (2.8) but also to (2.10), (2.11), (2.12) provided that a suitable bfs is found. In the following we show the results associated to (2.12).

Proposition 3. *The following elements:*

$$\mathbf{x}' = \begin{pmatrix}\boldsymbol{p}' & \boldsymbol{\xi}' & \boldsymbol{\zeta}' & \boldsymbol{\delta}_+{}' & \boldsymbol{\delta}_-{}' & \boldsymbol{\theta}_+{}' & \boldsymbol{\theta}_-{}'\end{pmatrix}'$$

$$\mathbf{a}' = \begin{pmatrix}\mathbf{0}' & \mathbf{1}' & \mathbf{0}' & \mathbf{0}' & \mathbf{0}' & \mathbf{0}' & \mathbf{0}'\end{pmatrix}'$$

$$\mathbf{c}' = \begin{pmatrix}\mathbf{0}' & \mathbf{0}' & \mathbf{1}' & \mathbf{0}' & \mathbf{0}' & \mathbf{0}' & \mathbf{0}'\end{pmatrix}'$$

$$\mathbf{A} = \begin{pmatrix} \mathbf{-I} & \mathbf{-I} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{I} & \mathbf{-I} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{\Sigma} & \mathbf{0} & \mathbf{D} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{-\Sigma} & \mathbf{0} & \mathbf{D} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \\ \mathbf{E} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix} \quad \mathbf{b} = \begin{pmatrix} \mathbf{H} \\ \mathbf{-H} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix}$$

with the block diagonal matrix $\mathbf{E} \in \mathbb{R}^{N \times ND}$, $\mathbf{E} = \text{diag}(\mathbf{1})$, $\boldsymbol{\xi}$, $\boldsymbol{\theta}_+, \boldsymbol{\theta}_- \in \mathbb{R}^{(N-1)D}$ and $\mathbf{1} \in \mathbb{R}^D$ *define (2.12) in the standard form (2.16) of a parametric LP.*

Proposition 4. *The set of indices* $\bar{\mathcal{B}}_0 = \mathcal{I}_1 \cup \mathcal{I}_2 \cup \mathcal{I}_3 \cup \mathcal{I}_4 \cup \mathcal{I}_5$ *with* $\mathcal{I}_1 = \{kD + 1 : k = 0, \ldots, N - 1\}$, $\mathcal{I}_2 = \{ND + k : k = 1, 2, \ldots, ND\}$, $\mathcal{I}_3 = \{2ND + N - 1 + kD + 1 : k = 0, 1, \ldots, N - 1\}$, $\mathcal{I}_4 = \{3ND + N - 1 + k : k = 1, 2 \ldots, ND\} \setminus \{3ND + N - 1 + kD + 1 :$

Algorithm 2 Mutiscale analysis in one-shot

Input: $\mathbf{H} = (\boldsymbol{h}'_1, \ldots, \boldsymbol{h}'_N)'$, $i = 0$.
Set (2.8) (or (2.12)) in standard form (2.16) according to Proposition 1 (Proposition 3).
Find an optimal bfs $\mathcal{B}_0$ for $\lambda_0 = \infty$ following Proposition 2 (Proposition 4).
while $\lambda_i \geq 0$
    Compute $\mathbf{x}^i$, with $\mathbf{x}^i_{\mathcal{B}_i} = \mathbf{A}^{-1}_{\mathcal{B}_i}\mathbf{b}$ and $\mathbf{x}^i_{\mathcal{N}_i} = \mathbf{0}$.
    $\bar{c}_j = c_j - \mathbf{c}_{\mathcal{B}_i}\mathbf{A}^{-1}_{\mathcal{B}_i}\mathbf{A}_j$ with $j \in \mathcal{N}_i$.
    $\bar{a}_j = a_j - \mathbf{a}_{\mathcal{B}_i}\mathbf{A}^{-1}_{\mathcal{B}_i}\mathbf{A}_j$ with $j \in \mathcal{N}_i$.
    $m = \arg\max_j\{-\frac{\bar{c}_j}{\bar{a}_j} : \bar{a}_j > 0\}$ (*entry index*)
    $\lambda_{i+1} = -\frac{\bar{c}_m}{\bar{a}_m}$
    $\mathbf{u} = \mathbf{A}^{-1}_{\mathcal{B}_i}\mathbf{A}_m$.
    if the support $I(\mathbf{u})$ is empty **then**
        return *problem is unbounded*
    $\ell = \arg$ lexico- $\min_t\{\frac{\mathbf{A}^i_t}{u_t} : t \in I(\mathbf{u})\}$ (*exit index*)
    Update $\mathcal{B}_{i+1} = \mathcal{B}_i \cup \{m\}\backslash\{\ell\}$
    Create the set $\mathcal{P}_i = \{\boldsymbol{p}^i_1, \ldots, \boldsymbol{p}^i_N\}$ extracting the corresponding coordinates from $\mathbf{x}^i$.
    $i \leftarrow i + 1$.
end
Output: The sets of prototypes $\mathcal{P}_1, \mathcal{P}_2, \ldots \mathcal{P}_{N_{prot}}$

$k = 0, 1 \ldots, N-1\}$, $\mathcal{I}_5 = \{4ND + N - 1 + k : k = 1, 2, \ldots, 2(N-1)D\}$, *individuates a bfs for (2.12).*

Proof. See Appendix B.

As a final remark we should note that in general even when the coefficients η_{ij} assume different values that in the case of temporal segmentation, (2.8) and (2.12) are also parametric LP problems and Algorithm 2 can be used for computing the entire solution path. However, in this cases (*e.g.* for nearest neighbor clustering) determining a suitable bfs $\mathcal{B}_0$ is more complex and we leave it to future works.

2.4 Experimental Results

2.4.1 Datasets and Experimental Setup

Experiments were conducted on five datasets, four of which are publicly available. The first dataset consists of a **Traffic** scene sequence. As the vehicles flow is controlled by traffic lights, different events occur at regular periods. The second video sequence depicts a **basketball match** and is taken from the APIDIS* website. The images are cropped to include only the basketball court and resized. The last three datasets

*http://www.apidis.org/Dataset/

TABLE 2.1: Details on datasets and experimental setup

	n^o frames	fps	n^o clips	frame size	patch size	D	clip length [s]
Traffic	6000	12	300	276×336	23×21	8	12
Basket	6000	23	100	320×368	16×16	16	3
Junction	90000	25	300	288×360	12×12	16	12
Roundabout	93500	25	311	288×360	12×12	16	12
Junction2	78000	25	312	288×360	12×12	16,24,30	10

TABLE 2.2: Proposed approaches tested in our experiments.

	L_1	*EMD-L_1-lin.*	*EMD-L_1-circ.*	*EMD-L_1-2D*
Formulation	(2.12)	(2.10)	(2.11)	deducible from (2.10)

Junction, **Roundabout**, **Junction2** are also available[†] (for the first two sequences, ground truth for two levels temporal segmentation is available; for the third one, we manually annotated a sequence of 80 clips at 2 and 3 levels, based on the traffic lights' changes). The videos depict some traffic scenes in London and have been extensively used in previous works [17, 18, 25, 45].

In this section, we first show temporal segmentation results obtained with EMD-L_1-*linear* (2.10); the other experiments are meant to test the proposed approach for nearest neighbor clustering. In the first case, *temporal segmentation* is obtained by setting in (2.10) $\eta_{ij} = 1$ if $i = j - 1$ and $\eta_{ij} = 0$ elsewhere; in the case of *clustering*, the nearest neighbor graph for prototype learning is computed based on histograms similarity, using EMD with L_1 ground distance. In all the experiments we found that $N_P = 3$ or $N_P = 4$ correspond to the best performance. A discussion about how to choose the values of α and β is reported in subsection 2.4.4. The value of λ changes in all the different experiments according to the required number of clusters. While for temporal segmentation Algorithm 2 can be used to obtain all possible prototypes at varying λ, for nearest neighbor clustering is necessary to test several λ to get the required number of clusters. More details about the datasets and our experimental setup are summarized in Table 2.1. The proposed algorithms are listed in Table 2.2 and are fully implemented in C++ using the publicly available libraries OpenCV for video processing and feature extraction and GLPK 4.2.1 (GNU Linear Programming Kit) as the backend linear programming solver. The code for solving problems (2.8), (2.10), (2.11) and (2.12) and the video showing our results are available online[‡].

[†]http://www.eecs.qmul.ac.uk/∼jianli/Dataset_List.html
[‡]http://disi.unitn.it/∼ zen/demo_emp.html

TABLE 2.3: Traffic dataset: temporal segmentation accuracy

EMD (2.8)	EMD-L_1-*linear*(2.10)	L_1(2.12)	Fused Lasso
83.2	82.4	72.5	68.7

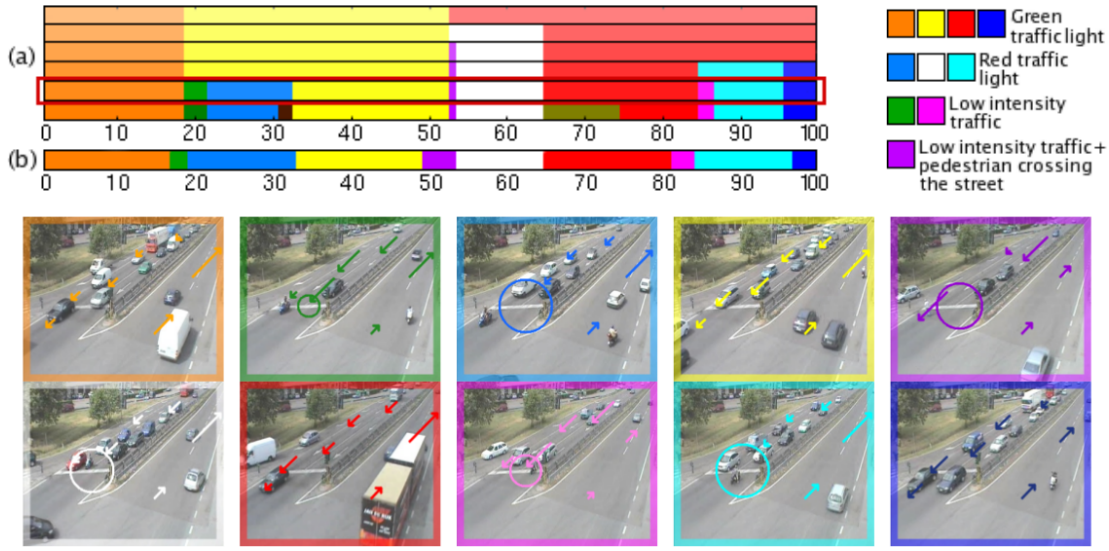


FIGURE 2.4: Traffic dataset. (a) Temporal segmentation results obtained varying λ with EMD-L_1(2.10). (b) Ground-truth and (top, right) corresponding legend. (Bottom) salient activities automatically extracted from the segmentation result highlighted in red.

2.4.2 Temporal Segmentation

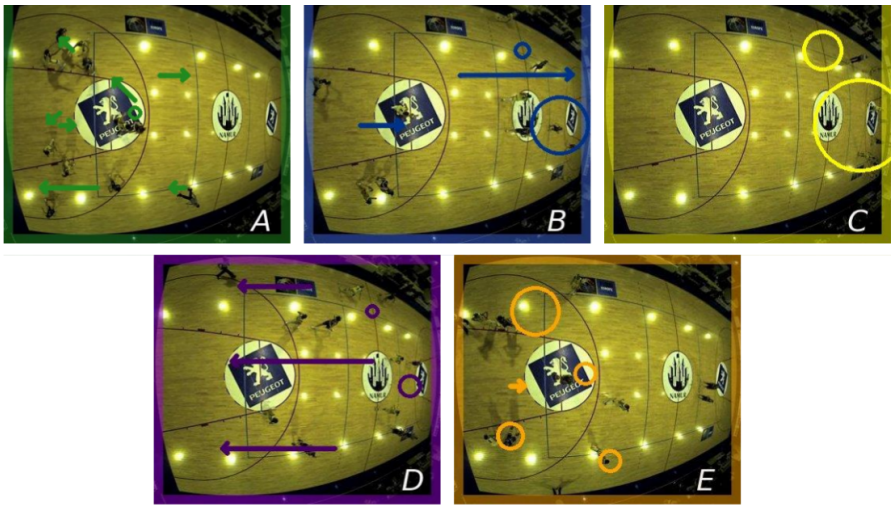
We demonstrate the effectiveness of the proposed temporal segmentation approach on the **Traffic** dataset. This sequence despite being short is interesting, as it corresponds to few cycles of the traffic lights status and it contains some interesting anomalous events. When applying temporal segmentation only the similarity among adjacent clips is considered. Therefore several prototypes corresponding to the same patterns (*e.g.* green traffic light) must be merged manually after learning. This supplementary phase may result annoying when dealing with long sequences (*e.g.* Junction, Roundabout). In these cases nearest neighbor clustering is preferred. For this reason we evaluate the performance of temporal segmentation results in term of correctly individuated break-points while for nearest neighbor clustering the accuracy is computed considering the percentage of correctly labeled clips. In the Traffic scene two main traffic flow patterns are distinguished: (i) two parallel flows when the traffic light is on green and (ii) vehicles stopped in the lane on the left when the traffic light is on red. Rare events also occur such as pedestrians crossing the street outside zebra crossing or vehicles making U-turns. Figure 2.4 shows the multi-scale segmentation obtained by solving EMD-L_1-*linear* (2.10)

for different values of λ. The temporal segmentation results with 10 clusters, obtained with $\lambda = 5$, is highlighted with a red frame. From each of the 10 clusters obtained we extract one frame, representative for the salient activities. As expected, clips with similar activity histograms are associated to the same cluster. Interestingly, we successfully detect the changes in vehicles flow triggered by the traffic lights. As shown in Fig. 2.4, the orange, yellow, red and blue clusters correspond to the activity of parallel vehicle flows (green traffic light), while the light blue, white and cyan clusters are associated to stationary vehicles (red traffic light). The green, violet and pink clusters are still associated to red traffic lights and, in particular, they represent the phase when the traffic queue begins, hence the traffic flow is characterized by low density.

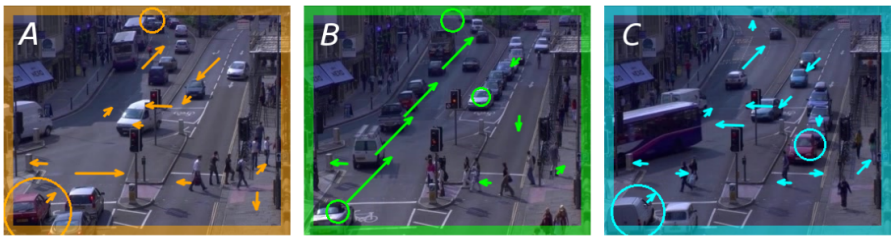
It is interesting to analyze the way clusters merge as λ increases. For example, the clusters associated to the same traffic light status but with different traffic density (*i.e.* pink and cyan, green and light blue) merge at the superior level. A visual inspection confirms that the segmentation results obtained with EMD distance are consistent with the human annotation (Fig. 2.4(b)). We manually annotated it. A quantitative comparison of the proposed methods (2.8) and (2.10) and bin-to-bin approaches (Fused lasso [34] and (2.12)) for the entire **Traffic** sequence is shown in Table 2.3. The performance is measured in terms of percentage of break points correctly individuated. The results clearly demonstrate that bin-to-bin distances are less powerful as they do not take into account similarity among atomic activities. It is worth noting that (2.10) can be considered as a good approximation of (2.8). An important observation concerns the computational cost of our multiscale analysis. As (2.8) is a parametric LP, *all* solutions (*i.e.* *all* possible prototypes) can be found with a slightly increased computational cost with respect to computing just one solution (corresponding to a fixed value of λ). Therefore, the speedup is huge. For example all possible prototypes associated to 100 clips can be computed in approximately 5 min whilst the solution for a single value of λ takes about 1 min.

2.4.3 Clustering

Discovering Salient Activities. We demonstrates that the proposed nearest neighbor clustering approach can be used to detect typical activities in various scenarios. For example for the **Basket** dataset five main activities are automatically identified: (A)



(a) Basket dataset



(b) Junction2 dataset

(c) Junction dataset

(d) Roundabout dataset

FIGURE 2.5: Salient activities extracted with our method. Circles and arrows represent static and dynamic atomic activities respectively (size is proportional to bin value)

when the yellow team is on defense and the blue team is trying to shot, (B) when the players are moving from the yellow team's court side to the blue team's side, (C) when the blue team is on the defense, (D) when the players are moving back towards the yellow team's side. Moreover, due to the asymmetric disposition of the camera with respect to

the basketball court, different phases of the match can be observed when players are in the yellow team's side, such as the case of free throws (E). A representative frame for each of the five activities automatically extracted solving (2.10) is shown in Fig. 2.5(a)

For the dataset **Junction2** we use our approaches for both two and three classes segmentation. The representative frames corresponding to the 3 clusters case automatically extracted are shown in Fig. 2.5(b) These three flow patterns are regulated by three traffic lights: flow (A) corresponds to red traffic light in the bottom left lane; flow (B) to red traffic light in the central lane; flow (C) to red traffic lights in the bottom left and in the right lane. Atomic activities corresponding to pedestrians crossing the main road are also individuated (see the small arrows in the lower part of the images).

For the **Junction** dataset (Fig. 2.5(c) by solving (2.10) or (2.11) we discover three main activities which correspond to different phases of the traffic flow: A) vertical flow and B) and C) respectively horizontal traffic flow from right to left and from left to right. These activities are also found in [17, 18, 46], with the difference that the cluster A is split in two different activities, corresponding to vertical flow with and without interleaved turning traffic. This division is less evident as it is confirmed by the transition behavior matrix in Fig.3.e in [17]. In fact, with our algorithm these patterns emerge when refining the analysis with more than three clusters. For the **Roundabout** dataset (Fig. 2.5(d)) two salient activities are discovered: they roughly correspond to the vertical (orange cluster) and the horizontal traffic flow (green cluster).

Comparison with Results in the Literature. We perform a quantitative comparison between our methods and PTMs. Table 2.4 shows the results (percentage of correctly labeled clips) obtained by applying our methods (2.10) and (2.11) to the **Basket** sequence compared to (2.12) and to pLSA with binary and tf-idf features representation. For pLSA clustering labels are obtained by taking the topic with larger probability. pLSA has been chosen as a baseline since it has been extensively used in previous works [24, 45]. We consider the results for 2 and 5 clusters. The ground truth is taken from the APIDIS website[§]. In the case of 2 clusters the ground truth is created by merging the activities A and E on one side, fusing B, C and D on the other. Table 2.4 confirms the advantages of EMD-based approaches w.r.t. competing methods. For example, in the

[§]We consider the timestamps of annotated events (*e.g.* 'Ball possession', 'Lost-ball', 'Free-throw', etc.) and added some missing information, *e.g.* the one representing a switch from events B to C or from D to A (Fig. 2.5(a)).

TABLE 2.4: Comparison of our approach with pLSA

	n^o clusters	EMD-L_1 *linear*(2.10)	EMD-L_1 *circular*(2.11)	L_1 (2.12)	pLSA	pLSA bin
Basket	2	**98.42**	**98.42**	**98.42**	94.15	92.25
	5	**90.84**	**90.84**	75.17	83.5	77.5
Junction2	2	**96.20**	93.67	93.67	93.67	86.08
	3	84.81	**86.08**	70.89	79.40	75.60

TABLE 2.5: Junction2 dataset: accuracy at varying number of atomic activities

n^{o}clusters	2			3		
n^{o}activities	30	24	16	30	24	16
k-means	88.83	**96.20**	94.41	68.24	67.05	59.73
L_1	93.67	**96.20**	**96.20**	70.89	69.20	70.89
EMD-L_1-*lin.*	**96.20**	**96.20**	86.08	**84.81**	55.70	56.96
EMD-L_1-*circ.*	93.67	**96.20**	**96.20**	**86.08**	68.35	70.89
EMD-L_1-*2D*	**96.20**	**96.20**	**96.20**	**89.87**	72.15	73.42

case of 5 clusters our methods outperforms pLSA with 7% in accuracy. We explain this with the fact that differently from (2.12) and pLSA, our approaches takes into account atomic activities similarity. Moreover, it is worth noting that pLSA results depend upon initialization conditions, as training relies on a non-convex problem. On the 2 clusters task there is no advantage on using EMD based methods with respect to using bin-to-bin clustering approach (2.12). We believe that in some easy tasks bin-to-bin distances may suffice.

Similar conclusions can be made for the dataset **Junction2** (see Table 2.4). Also in this case EMD-based approaches outperform L_1 clustering and pLSA for the most difficult task (3 clusters). Other interesting remarks can be made observing Table 2.5. Here the results obtained with all proposed approaches are compared at varying number of atomic activities. The table demonstrates that few atomic activities may not suffice for accurate segmentation. This is basically due to the fact that missing atomic activities hinder the recognition of high level behavior. For example for $D = 16$ the absence of the static atomic activities in upper left corner of the image inhibits the possibility to detect situations of traffic line (see Fig. 2.6). In these cases a $2D$ histogram representation with appropriate sorting compensates the decrease in accuracy. In this experiment we also report the results associated to *k-means* clustering as a baseline (Table 2.5). As expected, *ad-hoc* approaches as the ones we developed outperform standard clustering techniques.

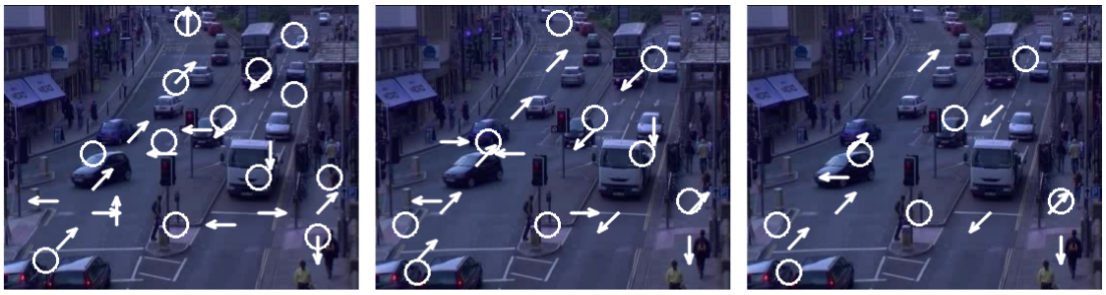


FIGURE 2.6: Junction2 dataset: different atomic activities extracted with (left) $D = 30$, (center) $D = 24$ and (right) $D = 16$.

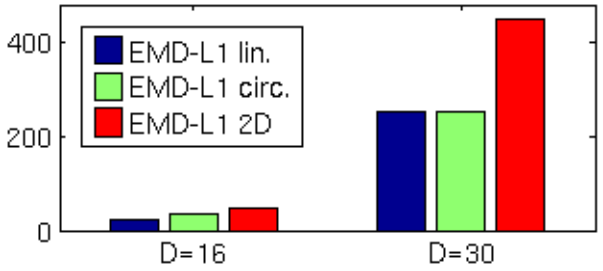


FIGURE 2.7: Junction2 dataset: average computational time (sec) for solving our learning problems at varying D.

In our experiments on the four datasets, we found that the EMD-L_1 with linear histograms and EMD with thresholded L_1 distance and circular histograms perform similarly (see Table 2.4 for the **Basket** and **Junction2** sequences) with a slightly better performance for the latter representation (Table 2.5). Therefore with our approach we did not found great benefits in using a thresholded ground distance opposite to what was reported in the previous works [43]. This is probably due to the fact that we do not simply compute the EMD between noisy histograms as in [43] but we use EMD as an objective function to calculate the set of prototypes.

An important consideration concerns the computational cost associated to our approaches. Figure 2.7 reports the average time (s) for solving the proposed optimization problems (3.5 GHz Intel Xeon machine). As expected the computational costs associated to prototype learning of 1D histograms are comparable, while a 2D representation implies an increased cost due to a larger number of flow variables.

Table 2.6 compares our approach with previously published results. In particular we consider the results reported in [18, 25]. We apply (2.10) and (2.11) on the same data (the datasets **Junction** and **Roundabout**) using the same clip size as [25]. Results reported in [18] are obtained using a slightly different settings, *i.e.* clip length= 3 sec and 6 clusters. We manually merged these clusters to directly compare with the ground truth in [25]. The corresponding temporal segmentation bars for the Junction dataset

TABLE 2.6: Comparison with previous works: clustering accuracy

	EMD-L_1 *linear*(2.10)	EMD-L_1 *circular*(2.11)	L_1 (2.12)	Standard pLSA[25]	Hierarchical pLSA[25]	DDP-HMM [18]
Junction	**92.31**	**92.31**	89.74	89.74	76.92	87.18
Roundabout	**86.40**	**86.40**	**86.40**	84.46	72.30	85.14

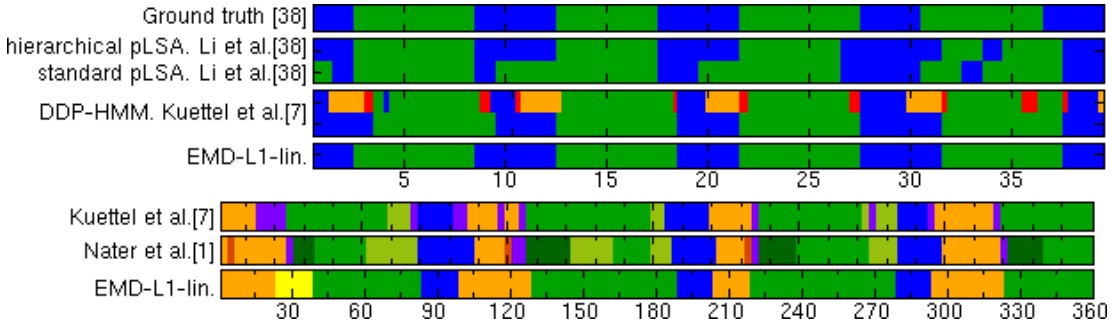


FIGURE 2.8: Junction dataset. Comparison with previous works.

are shown in Fig. 2.8(top). On both datasets the proposed algorithms outperforms DDP-HMM [18], pLSA and hierarchical pLSA [25] (the experimental setup is slightly different as in [25] a training/test approach is used). EMD-based clustering is also more accurate than prototype learning with L_1 distance (2.12). These results confirm the fact that higher clustering accuracy can be obtained by considering atomic activities similarity during the learning phase. In the case of the Junction dataset we also compare our approach with the results presented in [46] which correspond to the short sequence of 360 sec, between frame 9201 and 18200, segmented at 7 levels. These results do not refer to the same part of the sequence annotated in [25], so a quantitative comparison is not possible. A qualitative comparison between our approach and [18, 46] is provided in Fig. 2.8(bottom). As shown, the results of all three approaches are similar.

2.4.4 Ordering Atomic Activities

In this Section we present results demonstrating the validity of the proposed approach for sorting atomic activities. Table 2.7 proves the importance of choosing an appropriate order of atomic activities for EMD prototype learning: for all the datasets a random order of atomic activities entails a decrease in terms of accuracy. Figure 2.9 shows an example of atomic activities automatically sorted for the **Basket** and the **Junction2** datasets in the case of EMD-L_1 with circular histograms and thresholded ground distance. For Basket dataset, this order corresponds to the highest accuracy (90.84% in the 5 clusters case) and it is obtained for values $\alpha = \beta = 0.5$, *i.e.* considering both

TABLE 2.7: Clustering accuracy with and without sorting.

		Junction	Roundabout	Basket	Junction2
EMD-L_1-*lin.*	sorted	92.31	86.4	90.84	84.81
	unsorted	86.7	72.3	82	75.95

FIGURE 2.9: Automatically sorted atomic activities.

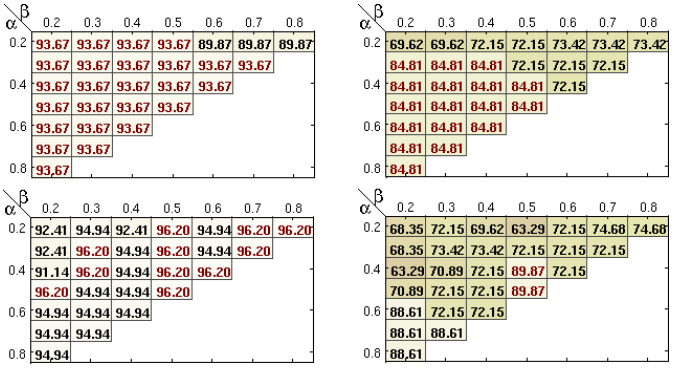


FIGURE 2.10: Junction2 dataset (D=30). Clustering accuracy for 2 (left) and 3 (right) clusters with different atomic activities orders using (top) EMD-L_1-lin. and (bottom) EMD-L_1-2D.

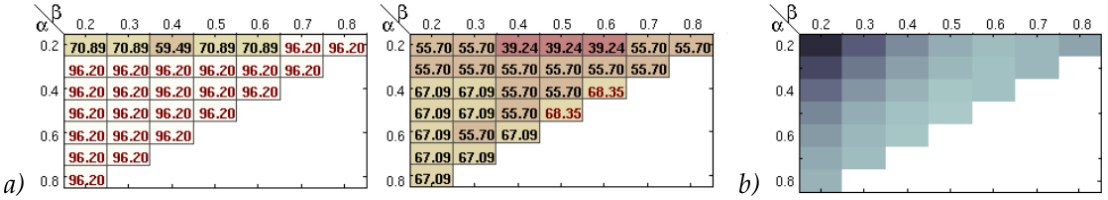


FIGURE 2.11: Junction2 dataset (D=24). **a)** Clustering accuracy for 2 (left) and 3 (right) clusters for different atomic activities orders using EMD-L_1 circular. **b)** Associated distortion matrix (higher is darker)

the motion and the position information when computing the optimal sorting. It is straightforward to observe that similar atomic activities are grouped (for example the first 5 activities correspond to zero motion). In this way atomic activities typically corresponding to the same cluster (*e.g.* number 0, 1 and 2 for the Free Throw) are close in the histogram representation.

FIGURE 2.12: Traffic dataset: anomaly (motorbike U-turn)

Figure 2.10 reports the performance of the proposed approaches for the **Junction2** dataset at varying values of the parameters α and β, *i.e.* for different sorting. The plots demonstrate that in general while for an easy task (2 clusters) almost all type of sorting produces good results (accuracy around 95%), when more clusters are required it is very important to take into account both the motion and the position information. Temporal correlation is less important. Similar results were also obtained for the other datasets. Therefore as a practical rule of thumb we set $\alpha = \beta = 0.5$. Interestingly, in most of the cases we observe a certain correlation between the values of distortions computed with Eqn. (2.14) and the clustering accuracy (see Fig. 2.11). Therefore looking at the distortion values can also be a valuable hint for sorting atomic activities.

2.4.5 Detecting Anomalous Patterns

By computing the MAS on an entire video sequence we detected some anomalous activities (persistent clusters of small size). In the case of the **Traffic** dataset an example of an unusual pattern is the violet cluster shown in Fig. 2.4 corresponding to a jaywalker. By looking at the multiscale segmentation in Fig. 2.4(a) it is evident that the violet cluster, opposite to the others, "survives" for several levels. This single clip cluster correctly obtains a high MAS score as it is associated to an anomalous activity. Another example of anomalous activity in this sequence is shown in Fig. 2.12. Here a motorbike makes a U-turn. This also corresponds to a single clip cluster which persist at several levels.

Figure 2.13 (top) shows some examples of anomalous activities found by MAS analysis (Fig. 2.13, bottom) for the dataset **Junction**. Anomalous activities corresponding to persistent small size clusters show the moments where the vertical traffic flows are interrupted as a pedestrian is crossing the street (clip 27) and a fireman truck is passing (clip 83). The last anomaly (clip 98) corresponds to a rare event where two large

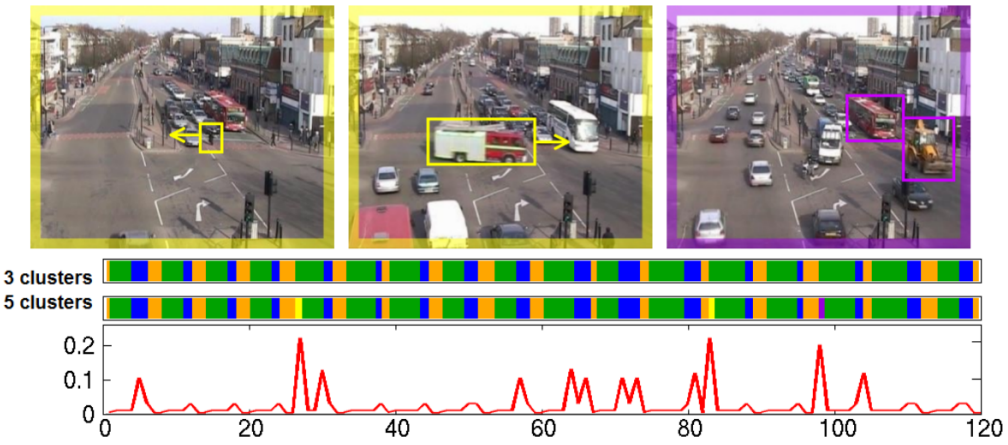


FIGURE 2.13: Junction dataset: detected anomalies (top) and the associated MAS plot (bottom).

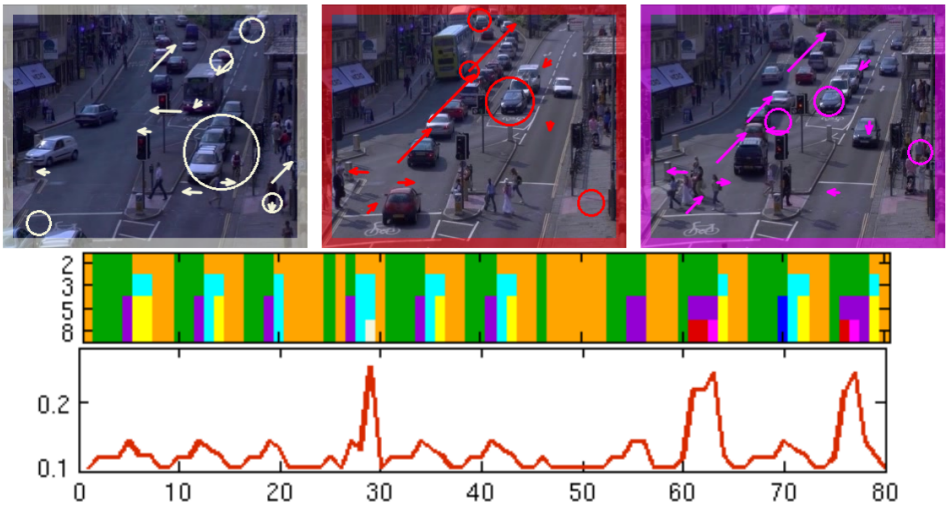


FIGURE 2.14: Junction2 dataset: detected anomalies (top) and the associated MAS plot (bottom).

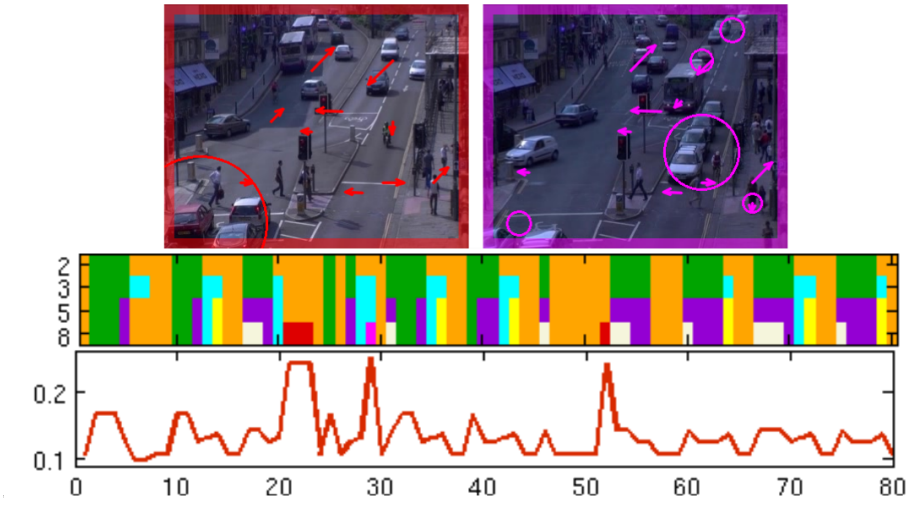


FIGURE 2.15: Junction2 dataset: detected anomalies (top) and the associated MAS plot (bottom) when atomic activities are not correctly sorted.

vehicles are passing at the same time. These results, similar to those in [25, 45, 46], confirm the validity of MAS analysis in finding anomalous events. In our experiments the MAS is computed considering $L = 9$ subsequent levels of segmentation. Figure 2.14 shows some anomalies detected for the **Junction2** dataset. Anomalies are due to an usual presence of a biker stopped next to the vehicles at the red traffic light (clip 28) and traffic jam on the left lane (clip 62,77). Finally Fig. 2.15 demonstrates that a good atomic activities sorting is also crucial for detecting anomalous patterns. Indeed, clip 28 is correctly individuated but clips 21-23 have a high MAS value even if they do not correspond to critical situations. In other words, in the case of incorrect non anomalous clips are indicated as anomalous, thus increasing the risk of false alarms.

2.5 Conclusions

We proposed a multiscale approach for discovering activity patterns in complex scenes. The main novelty of this work is the EMD prototype learning algorithm. By taking into account similarity amongst atomic activities, typical patterns can be extracted with improved accuracy with respect to previous approaches. The prototype learning algorithm has been presented in the context of dynamic scene analysis, but we believe that it could be successfully deployed in other tasks, such as action recognition.

In this work we considered the EMD approximation approach proposed in [13]. Recently, other methods [38, 47] have been proposed to speed-up the EMD distance calculation. These approaches are in general computationally more efficient than the one proposed in [13]. However, we chose Ling and Okada's approximation as it basically provides a simplification of the EMD definition proposing a LP with reduced flow variables. This LP can be easily embedded into our optimization framework and allows us to develop a Multiscale Analysis by Parametric LP theory. Moreover, the EMD wavelet approximation [38, 47] is especially convenient when the histogram size is larger than 200/300 bins. Differently, when very short histograms ($D \leq 50$) are considered as in this work, the EMD wavelet approach is not advantageous since the overall computational cost is dominated by the initial wavelets coefficients calculation.

Our experiments showed that the proposed approach is a valuable alternative to PTMs in the context of complex scene analysis. Differently from PTMs our approach takes into

account words similarity. However, it is worth noting that PTMs can be more versatile in applications when it is necessary to consider a large number of words, to learn the temporal dependencies among behaviors or to model the temporal information within the topics themselves. As an extension to this work, in Sec. 3 we show that by adopting the EMD approximation combined with Non-Negative Matrix Factorization (NMF) our clustering approach can be applied to other problems where high dimensional histograms are needed. In Chapter 3 we also show that is possible to effectively learn the ground distances by introducing a form of weak supervision.

Chapter 3

Discovering Patterns of Behaviors in Noisy Complex Video Scenes

3.1 Intuition

In Chapter 2 we proposed a novel approach which can successfully discover recurrent patters of behaviors in complex video scenes starting from simple low level visual cues. However, this method applies well when some specific assumptions hold, but there may be case scenarios where these assumptions do not hold completely. Inspired by the same motivations as in Chapter 2, in this chapter we proposed three approaches for discovering high level patterns of behaviors in complex scenes starting from the analysis of low level cues (e.g. motion, foreground information) which are specifically tailored to cope with the noise which is inherent in these cues. Besides, each of these methods is meant to better suit one of the specific cases described as follows.

- **High dimensionality of visual vocabulary**. The method presented in Chapter 2 suffers from scalability issues with respect to the vocabulary size. Indeed, at growing the number of atomic activities, the number of variables to be allocated in (2.8) and (2.10) also grows significantly, at a prohibitive cost in terms or RAM memory to be allocated. In Sec. 3.3 we present a novel method which is based on EMD as objective function and Non Negative Matrix factorization(NMF) as clustering method. Similarly to our previous work, the use of EMD allows to take

into consideration the similarity among atomic activities. The use of NMF allows higher scalability in terms of visual vocabulary size, however, at cost of losing the nice convex property which characterizes (2.8) and (2.10). Furthermore, in order to also cope with noise which is inherent in the low level features (e.g. foreground and local motion) extracted from complex and crowded scenes, a sparsity constraint on the computed basis matrix is imposed. This allows to filter out noise while leading to the identification of the most relevant elementary activities in a typical high level behavior. In Sec. 3.3.1 details are given on how the EMD-NMF approach can be solved efficiently via an alternate optimization approach. In Sec. 3.3.2 experimental results demonstrate that the proposed method yields similar or superior performance to state-of-the-art approaches.

- **Undefined *a priori* distance between atomic activities**. In Sec. 2.3.3, we show that atomic activities can be sorted by minimizing the distortion w.r.t. their original distances. In order to do so, the distance between atomic activities is defined in (2.9). However, this *a priori* definition may not effectively correspond to the distance between what these two atomic activities represent at a higher level in the scene. For example, two atomic activities may occur nearby in space and with opposite direction, but still be associated to the same high level activity, e.g. pedestrians crossing the street on a zebra crossing. Thus, it is desirable that the distance between this two activities is close to zero, because we want, for example, that having pedestrian flows towards one or the opposite direction on the same zebra crossing is indifferent at a prototype level. In Sec. 3.4 we propose a novel approach for learning high level activities prototypes which is based on Non-negative Matrix Factorization (NMF) as a clustering method and on Earth Mover's Distance (EMD) as a measure of reconstruction error. Differently from previous works on EMD matrix decomposition, we consider a semi-supervised learning setting and we also propose to learn the ground distance parameters. While few previous works have addressed the problem of ground distance computation, these methods do not learn simultaneously the optimal metric and the reconstruction matrices. We will show in Sec. 3.4.2 that our method allows not only to achieve state-of-the-art performance on video segmentation, but also to learn the relationship among elementary activities which characterize the high level events in the video scene. The effectiveness of the proposed approach is demonstrated both on synthetic data

and on a real world scenario, i.e. addressing the problem of complex video scene analysis in the context of video surveillance applications.

- **Long-term analysis of complex urban scenarios**. Most of the public datasets for the analysis of crowded dynamic scene are collected by recording imagery of public scenarios over a time range of maximum few hours, usually at daylight. As a consequence, typical patterns discovered are usually related to different traffic flows [3, 18, 24, 25]. Indeed, limiting the analysis over this short time period prevents the discovery of activities which occur at different moments of the day or at temporal frequencies of the order of days or weeks. In this work we propose a newly collected dataset and a method for the analysis of webcam imagery collected over a long term range. In these cases, due to the large amount of data storage required, the acquisition of imagery at a high framerate becomes infeasible. Instead, framerates are usually set to few frames per minutes or hours, thus impeding the extraction of low level cues such as local motion. In Sec. 3.5.2 we show that high level states such as traffic intensity can be monitored via exploiting only the foreground information extracted from a stream of webcam imagery. The analysis of typia and atypia required a robust method for background subtraction, which can perform expecially well in case of challenging lightening and weather conditions. For this purpose, we present a method based on sparse coding which outperforms state-of-the-art works on complex and crowded scenes (Sec. 3.5.1.2,3.5.2.3).

3.2 Related Work

EMD and NMF. The Non-negative Matrix Factorization algorithm aims to find two non-negative matrices whose product provides a good approximation to an initial matrix. While originally proposed to learn the parts of objects like human faces and text documents [48], in the last decades it has been applied to many other problems, such as action recognition [49], speech denoising [50], analysis of electromyographic signals [51] and blind source separation [52]. Typically NMF approaches adopt a bin-to-bin measure (*e.g.* L_2, Kullback Leibler divergence) to compute the reconstruction error. While this usually implies a simple optimization algorithm, the situations where the original matrix can only be obtained from complex deformations of some elementary signals cannot be modeled. To cope with this, the EMD-NMF algorithm is introduced in [36], where

a cross-bin measure, *i.e.* the Earth Mover's Distance, is adopted instead of bin-to-bin distances. Improved performance with respect to traditional NMF are shown in two computer vision tasks, *i.e.* texture descriptor estimation and face recognition. However, in [36] the ground distance values are kept fixed and are not optimized according to a discriminative criterion as proposed in this work.

EMD and ground distances. In order to overcome the main limitations of EMD, which are computational complexity and scalability, efficient versions of EMD have been proposed [13, 43, 47]. In some of them [13], the specific situation where the ground distance among histograms' bins is a linear function of the bin position is considered (*e.g.* $d_{ij} = |i-j|$ in EMD-L_1). In applications where different bin positions correspond to sorted elements in space [4] or in time [38] using a linear distance is a natural solution. Conversely, in situations where a histogram's bin corresponds to a word in a specific vocabulary (*e.g.* when the BoW paradigm is employed), the use of EMD-L_1 implies finding a reasonable words' order, according to which similar words are assigned to neighboring bin positions. Approaches for sorting have been proposed in literature [3, 53]. However they usually lead to a suboptimal solution, being the problem NP hard. Furthermore, as the initial distances are assigned based on L_1 or L_2 metrics, these may not necessarily reflect the discriminative ability of words. Therefore, by learning ground distances as proposed in this work we overcome these issues at the expenses of a slightly increased labeling efforts and computational cost in EMD calculation.

Simultaneous clustering and metric learning has been introduced in [54]. However, up to our knowledge, no previous works have considered these problems in the context of Earth Mover's Distance factorization. How learning the ground distance parameters affect EMD computation has been investigated in [16, 55]. In [55] an algorithm that learns the ground metric values using a training set of labeled histograms is proposed, overcoming the traditional approach that sets them based on a priori knowledge of the features. Wang *et al.* [16] also uses side information from triplets of samples (*i.e. must link* or *cannot link* constraints) to learn the cross-bin relationships, hence producing more accurate EMD values. However, in [16, 55] the ground metric parameters are learned in order to simply compute the EMD and not in the context of matrix factorization.

Long Term Video Analysis. Some of the key challenges associated with the analysis of data extracted from video over extended periods include requirements for storage and

long processing times. To reduce storage and make processing more efficient, studies of long-term video surveillance typically rely on video collected at low frame rates [29, 56]. However, low frame rates can make it difficult to reconstruct motion tracks for analyzing behavioral patterns. Another way to deal with the scale is to only collect data streams that are essentially static, *i.e.* the background appearance varies in time due to light, weather or seasonal changes but almost no foreground elements can be observed [56].

While there exists prior research on detecting anomalies in urban scenarios [29], these previous works focus on video analysis at frame level (*e.g.* extracting a pyramid of feature histograms). Further, there are hardly any approaches that attempt to distinguish the foreground elements from the background and to analyze their behavior separately. One of the exception is Abrams *et al.* [57] recorded a data set (LOST) with high frame rate from 17 cameras for over one year, in order to explore the changes in daily tracks. They record the same half hour each day, limiting the long-term analysis to a short interval in each day, and show that histograms of track density have a high-level interpretation w.r.t. natural human behavior. We show that this analysis can be done in a more efficient way and at a higher time granularity by exploiting the FG signal.

One of the core component of our work is a background subtraction module. In visual surveillance with static cameras, a BG subtraction method based on BG modeling is typically adopted. We considered the set of techniques described in recent surveys [58, 59] to select the most effective approach for our goals. The most widely used method was proposed by Stauffer and Grimson [60]. Among variations of this approach, [61] appears to be the most robust to the dynamics we face with the analysis of long-term streams, including *dynamic background*, *darkening*, and *noise during night time*. However, these techniques did not provide satisfactory results when applied on our long-term sequence (see Fig.3.24, 3.25 and supplementary material*). The results were fairly poor especially for the case of sudden light changes, a problem which is accentuated by the low frame acquisition rate. Also these approaches performed poorly during night because of low signal-to-noise ratio and presence of light reflections. A relevant set of works model the FG detection problem as a sparse signal recovery [62–67]. However, the methods rely on the assumption that the FG information is sparse. This latter assumption is not valid in our case where we routinely encounter crowded scenes (see Fig.3.21).

*http://disi.unitn.it/∼ zen/video/artemis13.avi

Our hypothesis in this work is that BG subtraction methods relying only on pixel-based analyses are not powerful enough. Instead, leveraging on more informative features (*e.g.* based on local structure) may be valuable. The use of local features like HOG [68] or texture has been proposed [69–71]. In this work, we show the effectiveness of using features learned via auto-encoders [72]. Such methods for learning features from data in an unsupervised manner have been applied successfully in a variety of fields (NLP, audio, computer vision, etc.), Most of the research in computer vision on using learned versus hand-designed features has focused on classification tasks like object recognition [73–75], image classification, [76, 77] or facial expression recognition [78], where the unsupervised phase of feature learning is combined with a supervised training phase of a classifier. To our knowledge, no work has yet explored the potential of using sparse features extracted at *patch*-level for the background subtraction task.

3.3 Earth's Mover Distance Non-negative Matrix Factorization

3.3.1 Method

In this Section, the proposed approach for extracting high level activities in complex scenes is presented. First, the features used to represent the short video clips are described (Sec. 3.3.1.1). Then, the proposed learning approach is illustrated (Sec. 3.3.1.2). For basic concepts about EMD and its variations, we refer the reader respectively to Sec. 2.3.2.1 and 2.3.2.2.

a)  b)  c)  d) 

FIGURE 3.1: Low level visual features used in our approach. (a) Original video frame and (b) associated foreground mask. (c) Trajectories (red) extracted with KLT tracker. (d) Trajectory snippets (red) and static pixels (blue) used to construct clip histograms.

3.3.1.1 Computing Clip Histograms

Similarly to previous works [1, 17, 18], we divide the video into short clips and we adopt a bag-of-words approach for computing clip histograms. First, we construct a codebook of *trajectons* as described in [79]. Feature trajectory snippets, *i.e.* sequences of (x_t,y_t) positions over time, are computed by cropping the features trajectories extracted using a KLT tracker [80]. Using a short video segment as training set, a codebook of trajectory snippets (the so-called trajectons) is computed by clustering the obtained trajectory snippets into a pre-specified number of clusters n_t. While in general standard k-means can be employed in this phase, in our specific application we manually selected the codebook ensuring that trajectories cover all the space of possible motion orientations. For the dynamic of the scene, in fact, a small codebook defining different motion orientation is more suitable to distinguish between the most relevant activities. This simple codebook corresponds to features more robust to noise than when considering optical flow vectors. In line with [80], we consider trajectory snippets formed by 10 positions in the trajectory. The subsequent phase consists in extracting low level features from the video and quantizing it according to the codebook generated. Specifically for each pixel we compute the foreground/background information using a simple dynamic Gaussian-Mixture background model as background subtraction algorithm [40]. We use KLT to compute trajectory snippets and assign them a label according to the nearest snippets in the codebook. The features extraction process is illustrated in Fig. 3.1. Then we divide the scene of interest in $n_x \times n_y$ patches, in order to take into account the location where the activities take place. We also divide the video into clips. A histogram counting the occurrences of trajectons labels is formed for each clip and each patch. Moreover, for each patch a further bin is used to account for static activities, *i.e.* pixels of foreground that do not belong to trajectons. The clip histogram $\boldsymbol{h}_i \in I\!R^{n_x \times n_y \times n_t}$ is obtained concatenating the patch histograms.

3.3.1.2 Discovering Activities with Sparse EMD Matrix Factorization

Given a training set of clip histograms $\mathcal{H} = \{\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots \boldsymbol{h}_N\}$, we model the task of discovering high level activities as the problem of finding a set of basis $\mathcal{P} = \{\boldsymbol{p}^1, \boldsymbol{p}^2, \ldots \boldsymbol{p}^K\}$, with $K << N$, and a matrix of mixing coefficients $\mathbf{W} = [\boldsymbol{w}_1\ \boldsymbol{w}_2 \ldots \boldsymbol{w}_N]$, $\boldsymbol{w}_i \in I\!R^K$, such that, for each clip, the weighted sum of the computed basis should be as close as

possible to the original clip histogram according to the Earth Mover's Distance. More formally the following optimization problem is formulated:

$$\min_{\boldsymbol{p}_k, \mathbf{W} \geq 0} \quad \sum_{i=1}^{N} \mathcal{D}_{EMD}(\boldsymbol{h}_i, \sum_k w_i^k \boldsymbol{p}^k) \tag{3.1}$$

$$\text{s.t.} \qquad \omega_m \leq \Omega(\boldsymbol{p}^k) \leq \omega_M, \quad \forall\ k = 1 \dots K \tag{3.2}$$

The imposed constraints force the computed basis to be sparse. To enforce sparsity, as in previous works on NMF [81, 82], we adopt the following measure:

$$\Omega(\mathbf{x}) = \frac{\sqrt{n} - \|\mathbf{x}\|_1 / \|\mathbf{x}\|_2}{\sqrt{n} - 1} \tag{3.3}$$

where $\mathbf{x} \in I\!R^n$. In practice the constraint (3.2) impose a lower and an upper bound (respectively ω_m and ω_M) to the level of sparsity of the computed prototypes.

By replacing the definition of EMD with L_1 ground distance (2.4) and $\Omega(\cdot)$ into (3.7), the following optimization problem must be solved:

$$\min_{p_q^k, w_i^k, f_{q,t}^i \geq 0} \quad \sum_{i=1}^{N} \sum_q \sum_{t \in \mathcal{N}(q)} f_{q,t}^i \tag{3.4}$$

$$\text{s.t.} \qquad \sum_{t \in \mathcal{N}(q)} f_{q,t}^i - \sum_{t \in \mathcal{N}(q)} f_{t,q}^i = h_i^q - \sum_k w_i^k p_q^k, \quad \forall q,\ \forall i$$

$$\sum_k w_i^k = 1, \quad \forall i \qquad \sum_q p_q^k = 1 \quad \forall k$$

$$\|\boldsymbol{p}^k\|_2 \leq \frac{1}{c_M} \mathbf{e}^T \boldsymbol{p}^k \quad \forall k \tag{3.5}$$

$$\frac{1}{c_m} \mathbf{e}^T \boldsymbol{p}^k \leq \|\boldsymbol{p}^k\|_2 \quad \forall k \tag{3.6}$$

where $c_M = \sqrt{Q} - \omega_M(\sqrt{Q} - 1)$ and $c_m = \sqrt{Q} - \omega_m(\sqrt{Q} - 1)$, $Q = n_x \times n_y \times n_t$ and $\mathbf{e} \in I\!R^Q$ is a vector of ones. The normalization constraints impose that each basis vector $\boldsymbol{p}^k$ and each column of the coefficient matrix $\mathbf{W}$ are normalized to sum one. This implies that $\sum_q \sum_k w_i^k p_k^q = 1,\ \forall\ i$, *i.e.* the reconstructed histograms are normalized to unit mass as required by EMD definition (2.4). The additional constraints (3.5) and (3.6) are imposed to force the basis to be sparse vectors.

The optimization problem (3.4) is not convex. However, to efficiently solve it, in this work we devise an approximate approach based on an alternate optimization scheme. We first consider (3.4) when constraints (3.5) and (3.6) are not imposed. In this case the problem (3.4) is still not convex. However if the coefficient matrix $\mathbf{W}$ is fixed, (3.4) is convex with respect to $p_q^k, f_{q,t}^i$. Similarly, with fixed basis vectors $\boldsymbol{p}^k$, (3.4) is convex

Algorithm 3 EMD Clustering

1: **Input**: Original clips histograms $\mathcal{H} = \{\boldsymbol{h}_1,\ \boldsymbol{h}_2, \dots \boldsymbol{h}_N\}$.
2: Initialize $\mathbf{W} = [\boldsymbol{w}_1\ \boldsymbol{w}_2 \dots \boldsymbol{w}_N]$ with positive random values.
3: Normalize the columns of $\mathbf{W}$ such that $\sum_k w_i^k = 1,\ \forall i = 1, \dots N$.
4: **while** not converged
5: Solve (3.4) s.t. (3.5) and (3.6) with respect to $\boldsymbol{p}^k, \boldsymbol{f}$ using Algorithm 4.
6: Solve the LP (3.4) with respect to $\mathbf{W}, \boldsymbol{f}$.
7: **end**
8: **Output:** $\mathbf{W}$, $\boldsymbol{p}^k\ \forall k$.

with respect to $w_i^k, f_{q,t}^i$. To solve it, an alternate optimization scheme can be devised where each single optimization problem reduces to a LP. This approach, which turns out to be a special case of the algorithm proposed in [36], can be shown to converge to a local minimum.

If the constraints (3.5) are also considered, the optimization problem (3.4) can still be solved with an alternate optimization scheme and, in particular, as a sequence of convex optimization problems. Solving with respect to $w_i^k, f_{q,t}^i$ with variables p_q^k fixed is still a LP, while solving with respect to $p_q^k, f_{q,t}^i$ having $\mathbf{W}$ fixed is a Second Order Cone Programming (SOCP) which can be solved efficiently with standard solvers. However, when the constraints (3.6) are also considered, solving (3.4) with respect to $p_q^k, f_{q,t}^i$ and w_i^k fixed is not convex anymore. Therefore, inspired by previous works on NMF [81], we adopt an approximate technique to solve it. The approach is based on the Tangent Plane Constraint (TPC) method [83] and basically consists in approximating the non convex cone constraints by linear constraints and specifically by tangent plane constraints.

The algorithms we develop for solving (3.4) are shown in Algorithm 3 and Algorithm 4. In particular the alternate optimization approach used to solve (3.4) is illustrated in Algorithm 3. Step 5 of Algorithm 3 consists in solving (3.4) subject to (3.5) and (3.6) with the TPC method. The TPC method is illustrated in Algorithm 4.

To solve the proposed optimization problem we adopt some practical solutions which reduce the computational cost of our approach and then makes it more appealing to large scale computer vision applications. First of all we note that the convex constraints (3.5) are not particularly important in order to guarantee sparse solutions. In fact, rather than imposing an upper bound on the maximum level of sparsity, it is much more important to guarantee a minimum level of sparsity. This implies that in practice we can omit convex constraints (3.5). This is of paramount importance in practical applications

Algorithm 4 Algorithm for computing sparse prototypes

Input: Original clips histograms $\mathcal{H} = \{\boldsymbol{h}_1, \boldsymbol{h}_2, \dots \boldsymbol{h}_N\}$ and coefficent matrix $\mathbf{W}$. The parameters ω_m and ω_M specifying the desired sparsity levels.
  Compute $c_M = \sqrt{Q} - \omega_M(\sqrt{Q} - 1)$ and $c_m = \sqrt{Q} - \omega_m(\sqrt{Q} - 1)$.
  Solve (3.4) s.t. (3.5) with respect to $\boldsymbol{p}^k, \boldsymbol{f}$.
  Initialize the index set of violated constraints $\mathcal{V}^0 = \emptyset$.
  Set $t = 0$.
  while not converged
    $\bar{\boldsymbol{p}}^k = \boldsymbol{p}^k, \forall k$.
    Find $\bar{\boldsymbol{p}}^k$ violating (3.6); update $\mathcal{V}^{t+1} = \mathcal{V}^t \cup \{r : r = 1, \dots, K, \frac{1}{c_m}\mathbf{e}^T\boldsymbol{p}^r \geq \|\boldsymbol{p}^r\|_2\}$
    $\forall r \in \mathcal{V}^{t+1}$ compute the projection $\bar{\pi}_r = \pi(\bar{\boldsymbol{p}}^r)$ as shown in [82].
    $\forall r \in \mathcal{V}^{t+1}$ compute the tangent plane $\boldsymbol{t}_{r,\bar{\pi}}^{t+1}$ to cone (3.6) in $\bar{\pi}_r$
    Solve (3.4) s.t. (3.5) and to the tangent plane constraints $(\boldsymbol{p}^r)^T\boldsymbol{t}_{r,\bar{\pi}}^{t+1} \geq 0, \ \forall r \in \mathcal{V}^{t+1}$ with respect to $\boldsymbol{p}^k, \boldsymbol{f}$.
    $t = t + 1$.
  end
Output: $\boldsymbol{p}^k, \forall k = 1, \dots, K$.

since the sequence of SOCP problem in Algorithm 4 (Step 3 and Step 4) reduce to a sequence of efficient LP problems. In alternative, as for the constraints (3.6), also in case of (3.5) tangent plane constraints can be devised. Still, the overall optimization problem reduces to a LP. In our experiments we used the former solutions ($\omega_M = 1$).

While the TPC method is guaranteed to converge (*i.e.* Algorithm 4 always converges) [83] the alternate optimization problem in Algorithm 3 is not guaranteed to converge. For this reason in [81], in case of TPC applied to NMF, a more robust but slower approach is proposed. While we also cannot prove the convergence of Algorithm 3 when using TPC method in our experiments we did not observe problems of convergence (Fig. 3.5).

While our approach can be generally applied to several types of histogram data, for computational efficiency reasons in our experiments we consider two-dimensional histograms obtained by reshaping the clip histograms as $\boldsymbol{h}_i \in \mathbb{R}^{n_x \times n_y \times n_t}$. In this way the EMD objective function operates on a grid as neighborhood structure, where neighboring bins in a histogram mostly corresponds to the same features (*e.g.* same trajectons) computed in neighboring patches.

TABLE 3.1: Details on the datasets and the experimental setup

	n^o frames	fps	video duration	frame size	$n_x \times n_y \times n_t$	Q	clip duration	n^o clips
Junction	90000	25	60'	288×360	8×6×9	432	12"	300
Junction2	78000	25	52'	288×360	8×6×9	432	12"	260
Roundabout	93500	25	62'	288×360	12×9×9	972	12"	311

3.3.2 Experimental Results

3.3.2.1 Datasets and Experimental Setup

Experiments were conducted on three publicly available datasets collected from researchers of Queen Mary University, namely **Junction**, **Junction2** and **Roundabout**. The videos depict some complex traffic scenes in London and have been extensively used in previous works [1, 16–18, 25]. The ground truth corresponding to activities found by a human annotator are also publicly available[†]. To compare our approach with state-of-the-art methods [1, 18] we also use the code and the results made available by other research groups[‡,§]. Our method is implemented in C++ using the publicly available library OpenCV for the video processing and feature extraction parts while MATLAB is employed for Algorithm 3 and 4. More details about the datasets used and our experimental setup are summarized in Table 3.1.

3.3.2.2 Discovering High Level Activities

The first series of experiments is aimed to demonstrate the ability of the proposed approach to extract high level activities by selecting the most significant elementary features in the scene. Figure 3.2 depicts the high level activity patterns computed with our approach for the Junction dataset. These three main patterns correspond to vertical traffic flow, horizontal flow from left to right and from right to left. In the same figures, the $n_t + 1$ elementary features are plot in different colors: green circles correspond to static activities and the other colors identify the n_t different trajectons, whose main direction is indicated by arrows. Also, the intensity of each elementary feature is represented by N_e colored patches that are plotted with a Gaussian distribution around the patch centroid (i, j). The number N_e is proportional to $p_k^{i,j,t}$.

[†]http://www.eecs.qmul.ac.uk/∼ jianli/Dataset_List.html
[‡]http://disi.unitn.it/∼ zen
[§]http://www.vision.ee.ethz.ch/∼ calvin/publications.html

a)
b)

FIGURE 3.2: Junction dataset. High level activities automatically extracted with our approach at different levels of sparsity (a) $\omega_m = 0.0$, (b) $\omega_m = 0.9$

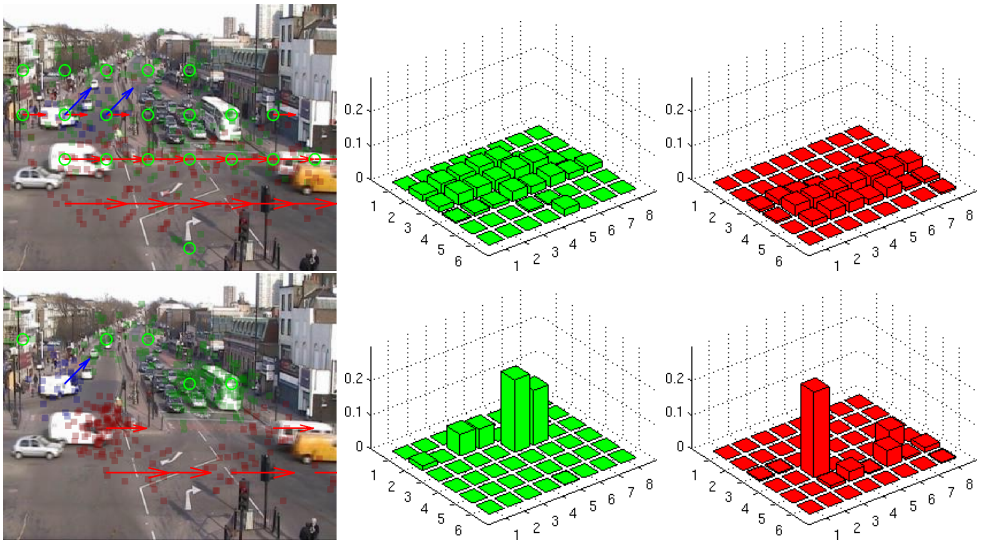


FIGURE 3.3: Effect of combining EMD with sparsity constraints, shown on Junction dataset. Prototype obtained with our method setting (top) $\omega_m = 0.0$ and (bottom) $\omega_m = 0.9$. The 2D histogram is shown for zero motion and for rightward motion elementary features (drawn respectively in green and red).

Varying the required minimum sparsity level, and specifically with ω_m close to one, only few elementary features are active in the final prototype representation. Furthermore a grouping effect, which must be ascribed to the use of EMD as objective function, is observed, as elementary features in adjacent regions tend to be active or not active together. The effect of sparse grouping activities can also be observed in Figure 3.3.

Similar observations can be made in case of the Roundabout dataset (Fig. 3.4) where six main activities are extracted. In details, the yellow and light/dark green activities

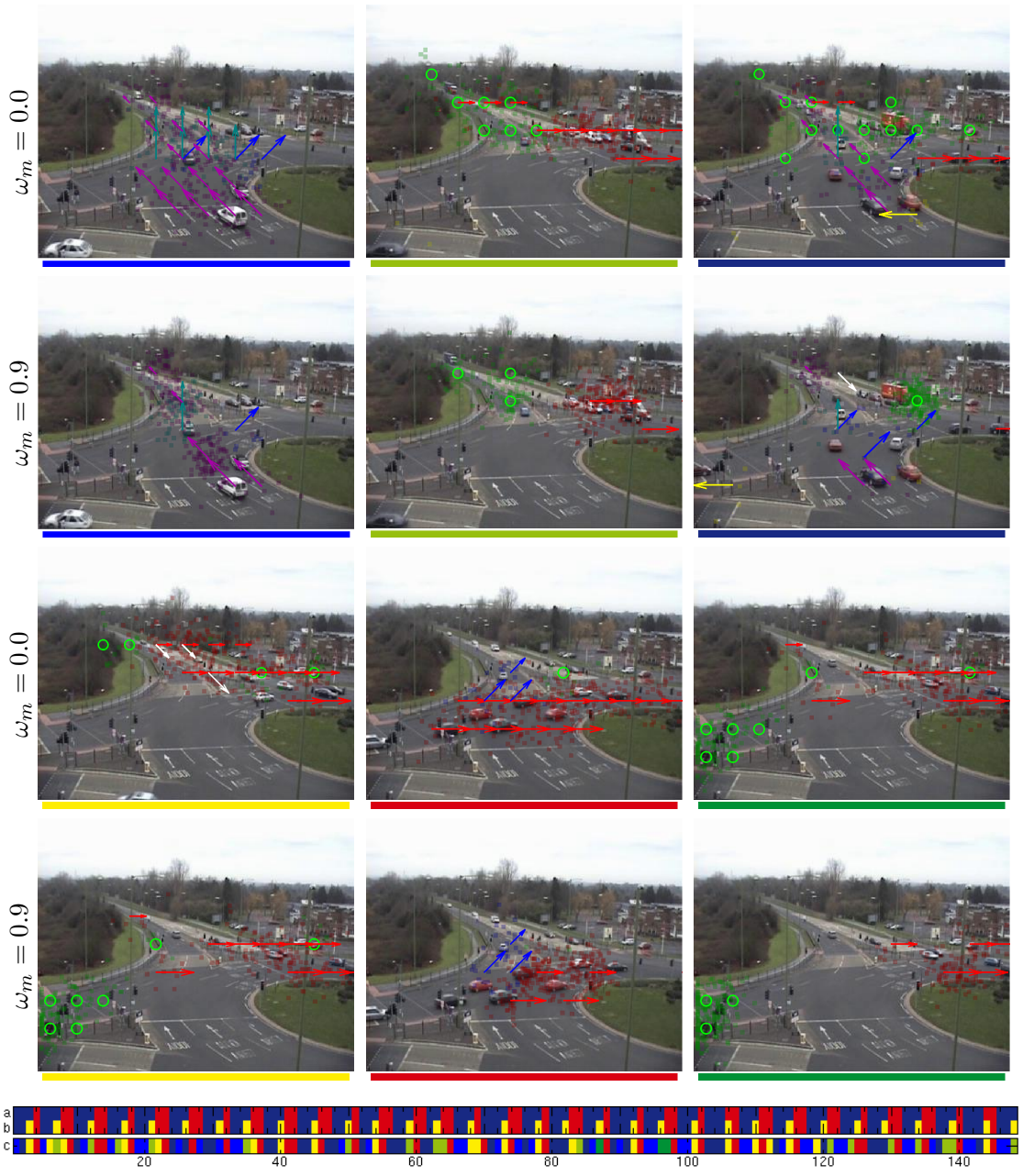


FIGURE 3.4: Roundabout dataset. (Top) High level activities automatically extracted with our approach at different levels of sparsity $\omega_m = 0.0$ and $\omega_m = 0.9$. (Bottom) Temporal bars: (a) GT [25] (b) GT considering three classes set by the authors and (c) temporal segmentation obtained with our approach.

correspond to the same higher level activity (top-right traffic lights on green) but at different traffic flow intensity. The blue and red activities correspond, respectively, to central-bottom and left traffic lights on green.

Table 3.2 depicts the clustering accuracy obtained by varying ω_m for the three datasets

TABLE 3.2: Clustering accuracy at varying sparsity level ω_m.

ω_m	0	0.1	0.3	0.5	0.7	0.9
Junction2 (48 clips)	89.58%	89.58%	89.58%	89.58%	**91.67%**	89.58%
Roundabout (60 clips)	88.33%	88.33%	**90.00%**	**90.00%**	**90.00%**	**90.00%**
Junction (39 clips)	**89.74%**	**89.74%**	**89.74%**	**89.74%**	**89.74%**	84.62%

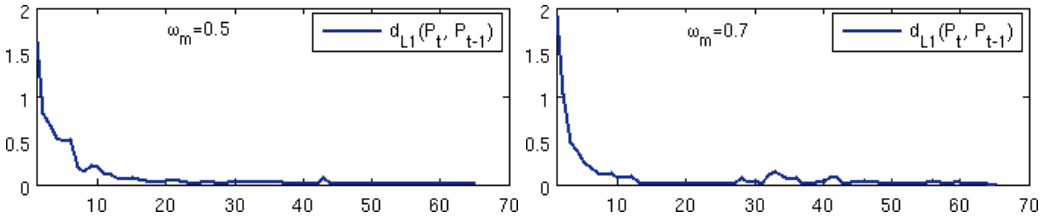


FIGURE 3.5: Junction dataset. Convergence analysis of our approach for different levels of sparsity (left) $\omega_m = 0.5$ and (right) $\omega_m = 0.7$.

considered. The results correspond to a two clusters groundtruth segmentation. Imposing sparsity constraints in the learning process is important for the semantic interpretation of video contents, and our experiments demonstrate that this does not negatively affect the accuracy. In some cases, when a high degree of sparsity ($\omega_m = 0.7$) is imposed, the performance can also be better. This can be ascribed to the beneficial effect of sparsity constraints in filtering out noisy features. In few cases instead, for severe level of sparsity ($\omega_m = 0.9$) the accuracy can slighly degrade. This is probably due to the loss of some details that could be useful for some classes' discrimination.

3.3.2.3 Convergence

As discussed in Section 3.3.1.2, Algorithm 3 is not guaranteed to converge when the Tangent Plane Constraint method [81, 83] is adopted. However, in our experimental results we mostly observed a convergent behavior. Figure 3.5 depicts two examples of convergence for the experiments conducted on the Junction dataset for $\omega_m = 0.5$ and $\omega_m = 0.7$; specifically the value shown is the L_1 distance computed between successive vector bases $\boldsymbol{p}_k^t$ and $\boldsymbol{p}_k^{t-1}$, at each iteration t. Some convergence issues were observed for values of ω_m close to 1. However these situations are of less practical utility as the best clustering accuracy is typically obtained for $\omega_m < 0.9$.

TABLE 3.3: Comparison with previous approaches: clustering accuracy

	std pLSA [25]	hrc pLSA [25]	DDP-HMM [18]	EMP [1]	our approach ($\omega_m = 0.7$)
Junction	89.74%	76.92%	87.18%	**92.31**%	89.74%
Roundabout (60 clips)	81.67%	75.00%	85.00%	86.67%	**90.00**%
Roundabout (148 clips)	84.46%	72.30%	85.14%	**86.40**%	*85.81*%

3.3.2.4 Comparison with Previous Works

In this Subsection we report some results aimed at comparing the proposed approach with previous methods [1, 17, 18, 25].

Temporal Segmentation. We first consider the datasets Junction and Roundabout, as for these videos a ground truth annotation with two classes (horizontal and vertical traffic flows) is provided in [25]. However, it is easy to observe that the natural classes of traffic flows are more than two. In particular, for the Roundabout dataset, this is due to the presence of more that two traffic lights regulating the vehicles' flow and to varying traffic flows intensity (*e.g.* traffic light is on green but there are no vehicles in the lane). Therefore Kuettel *et al.* [18] consider a temporal segmentation with $K = 6$. Moreover they use clips of 3 sec length instead of 12 sec as related works. In our experiments, we also show results obtained with $K = 6$. In Fig. 3.6 the results obtained with the different methods are compared. Specifically the segmentation computed with Probabilistic Latent Semantic Analysis (PLSA) e hierarchical PLSA [25], Dependent Dirichlet Process Hidden Markov Model (DDP-HMM) [18], Earth Mover's Prototypes (EMP) [1] and our approach are compared. As shown in the plot, all the approaches obtained consistent results with respect to ground truth annotation. Similar comparative results are also reported for the Junction (Fig. 3.7) and the Junction2 (Fig. 3.8) datasets. For these datasets, a qualitative comparison with the work in [17] is also possible, as our approach is able to extract the same recurrent activities shown in [17]. Note that for the Junction2 dataset only the results provided by [18] are available. A quantitative comparison between our approach and the methods [1, 18, 25] is also provided in Table 3.3.

Observing the first two rows of the table it is evident that our approach outperforms previous methods in the Roundabout dataset, while it is the second best for Junction. The last row in the table shows the segmentation results for a longer sequence of the Roundabout dataset. In this sequence the best results are obtained by the approach proposed in [1]. However it is worth noting that these results correspond to a different

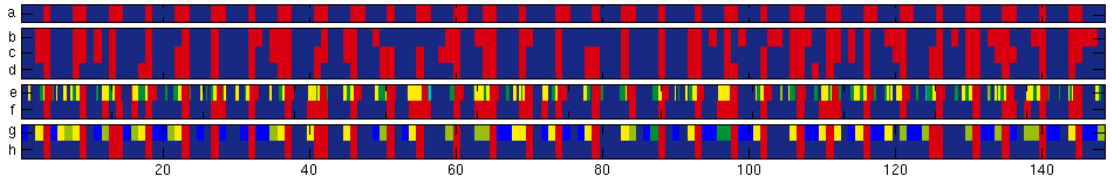


FIGURE 3.6: Roundabout dataset. (a) Ground truth annotation [25] and temporal segmentation results obtained with (b,c) standard and hierarchical pLSA [25], (d) EMP [1], (e,f) HDP-HMM [18] and (g,h) our approach.

FIGURE 3.7: Junction dataset. (a) Ground truth annotation [25] and temporal segmentation results obtained with (b,c) standard and hierarchical pLSA [25], (d) EMP [1], (e,f) DDP-HMM [18] and (g,h) our method. (Left) extracted high level activities $\omega_m = 0.9$.

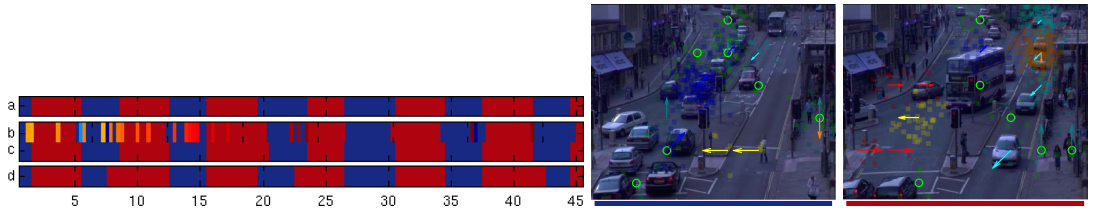


FIGURE 3.8: Junction2 dataset. (a) Ground truth annotation and temporal segmentation results obtained with (b) DDP-HMM [18] and (c) our method. (left) extracted high level activities with $\omega_m = 0.9$.

experimental set-up, as in [1] all the 148 clips are used as training set, while, similarly to [25] we consider a more challenging task and we train only on 60 clips and use the remaining clips as test set. In these experimental conditions we outperform previous methods.

Computational Cost and Comparison with EMP. In this section we compare the proposed method EMD-NMF and EMP (Ch. 2) in terms of computational cost. It worth noting that the features we used for EMD-NMF differ from the one we used for EMP. In particular, we could not test the method EMP with the features we used with EMD-NMF because the algorithm EMP does not scale with long histograms. Long histograms can be more suitable in case of EMD learning, as the similarity among bins is naturally imposed by the patch division structure. This is different from EMP, where an elementary activity order needs to be established to create clip histograms. However, in order to compare our approach with EMP, we use the same dense histogram representation as in Ch. 2.

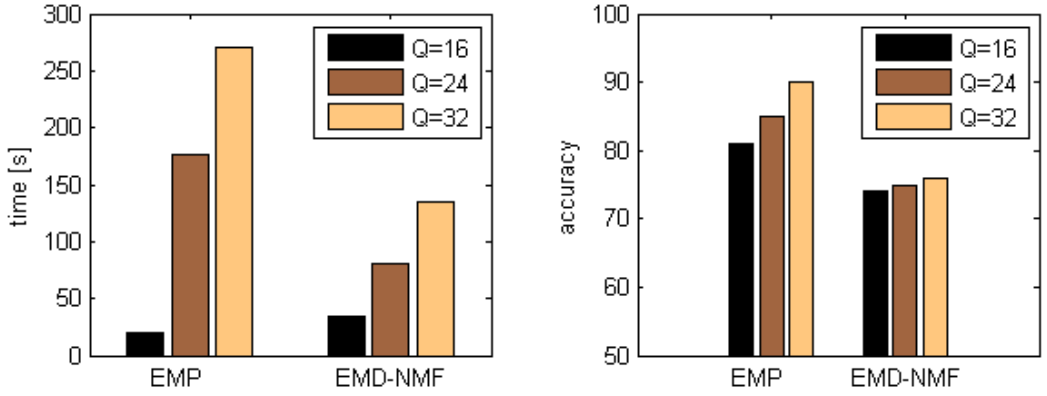


FIGURE 3.9: Junction2 dataset. Comparison on clustering dense histograms data at varying number of atomic activities Q. (Left) EMP and (right) EMD-NMF.

FIGURE 3.10: Junction dataset. (Left) Anomaly score. (Right) Representative frames extracted from the detected anomalous clips: (4,27) interruption of vertical traffic flow due to a fire engine passing, (9) leftward horizontal and (15) vertical flow, both interleaved with rightward horizontal traffic.

We compute one-dimensional histograms where each bin represents an atomic activity (in Ch. 2 an atomic activity consists in a specific motion pattern occurring in a specific image region). Atomic activities must be manually sorted. In this work we consider five different atomic activity orders. Our results are the average of these five runs. Figure 3.9 shows the results of our comparison. From the plots it is evident that, when histograms dimension Q increases, our approach is much more scalable. On the other hand, as expected, our method has modest performance in terms of accuracy. Our best results are obtained for $Q = 32$ and correspond to an accuracy equal to 75%. On the same data the algorithm in [1] reaches an accuracy of 92%. However, as demonstrated by Table 3.2, similar performance (91.67%) can be obtained with our approach when a sparse histogram representation is adopted.

Anomaly Detection. In this paragraph we briefly show that our approach can be used to identify anomalous and rare activities. To this aim the mixing coefficients $\mathbf{W}$ can be analyzed. Given a clip histogram $\boldsymbol{h}_i$ and the associated weights $\boldsymbol{w}_i$, we consider the corresponding activity as rare if it cannot be explained by the computed basis $\boldsymbol{p}_k$. This practically means that none of the w_i^k is close to one, *i.e.* the standard deviation $\sigma_{\boldsymbol{w}_i}$ of the coefficients w_i^k is small. With this intuition, $\sigma_{\boldsymbol{w}_i}$ can be used as anomaly

score. The anomaly score computed for the Junction dataset is shown in Fig. 3.10. Negative peaks identifying the anomalous clips are highlighted in green. Clips 4 and 27 correspond to the interruption of traffic flow due to a fire engine passing, Clips 9 is and 15 are anomalous as they are associated, respectively, to leftward horizontal flow and to vertical traffic, but they are also interleaved with rightward horizontal flow. These results are similar to those in previous works [1, 17] and correspond to the anomalies indicated in the ground truth.

3.4 Simultaneous Ground Metric Learning and Matrix Factorization

3.4.1 Method

3.4.1.1 EMD-NMF with Ground Metric Learning

We are given a training set $\mathcal{H} = \{\boldsymbol{h}_i\}_{i=1}^{N}$, $\boldsymbol{h}_i \in I\!R^M$ of normalized histograms and a small set $\mathcal{H}_s = \{(\boldsymbol{h}_i^s, \boldsymbol{h}_j^s, y_{ij}^s)\}_{i,j=1}^{N_s}$, of pairs of histograms $\boldsymbol{h}_i^s, \boldsymbol{h}_j^s \in I\!R^M$ and associated label $y_{ij}^s \in \{1, -1\}$ indicating if the histograms belong to the same or to a different class. From the set $\mathcal{H}$ we construct the matrix $\mathbf{H} = [\boldsymbol{h}_1\ \boldsymbol{h}_2 \dots \boldsymbol{h}_N]$, $\mathbf{H} \in I\!R^{M \times N}$. We are interested in decomposing $\mathbf{H}$ finding a set of basis $\mathcal{P} = \{\boldsymbol{p}^1,\ \boldsymbol{p}^2,\ \dots \boldsymbol{p}^K\}$, with $K << N$, $\boldsymbol{p}^k \in I\!R^M$ and a matrix of mixing coefficients $\mathbf{W} = [\boldsymbol{w}_1\ \boldsymbol{w}_2 \dots \boldsymbol{w}_N]$, $\boldsymbol{w}_i \in I\!R^K$, such that the weighted sum of the computed basis should be as close as possible to the original histograms according to Earth Mover's Distance. We also want to find the optimal ground distance parameters $\boldsymbol{d} \in I\!R^{M \times M}$ imposing that histograms of different classes $(\boldsymbol{h}_i^s, \boldsymbol{h}_j^s)$ in $\mathcal{H}_s$ should be more distant than histograms of the same class $(\boldsymbol{h}_l^s, \boldsymbol{h}_m^s)$. The following optimization problem is formulated:

$$\begin{aligned} \min_{\boldsymbol{p}^k, \mathbf{W}, \boldsymbol{d}} \quad & \|\boldsymbol{d}\|_F^2 + \lambda_1 \sum_{i=1}^{N} \mathcal{D}_{\boldsymbol{d}}(\boldsymbol{h}_i, \sum_{k=1}^{K} w_i^k \boldsymbol{p}^k) + \lambda_2 \sum_{ijlm} \xi_{ijlm} \\ & \mathcal{D}_{\boldsymbol{d}}(\boldsymbol{h}_i^s, \boldsymbol{h}_j^s) - \mathcal{D}_{\boldsymbol{d}}(\boldsymbol{h}_l^s, \boldsymbol{h}_m^s) \geq 1 - \xi_{ijlm} \quad \forall i, j, l, m \\ & \boldsymbol{p}^k \in \mathcal{F}, \mathbf{W} \geq 0, \boldsymbol{d} \in \mathcal{D} \end{aligned} \tag{3.7}$$

where $\mathcal{F} = \{\boldsymbol{p}^k \in I\!R^M : \sum_q p_q^k = 1,\ p_q^k \geq 0\}$ and $\mathcal{D} = \{\boldsymbol{d} \in I\!R^{M \times M} : d_{qt} \geq 0,\ d_{qt} = d_{tq},\ d_{qq} = 0\}$. In practice the feasible set $\mathcal{F}$ indicates that the set of the chosen basis vectors should be histograms normalized to unit mass, while the set $\mathcal{D}$ indicates that the matrix of ground distance parameters should be symmetric and with all the elements

equal or greater than zeros with the exception of the elements on the diagonal which are equal to zero. This choice corresponds to defining a valid transportation problem in the form (4.14).

3.4.1.2 Optimization

The optimization problem (3.7) is non convex. To solve it we adopt an alternate optimization approach and solve separately for $\boldsymbol{p}^k, \mathbf{W}, \boldsymbol{d}$ considering a sequence of convex optimization problems. In practice, at every step the optimal values of the flow vectors in the EMD definition must also be computed. In the following we describe the proposed optimization algorithm.

Initialization. Given $\mathcal{H}$, $\mathcal{H}_s$ initialize $\mathbf{W}$, $\boldsymbol{d}$. The initialization of $\mathbf{W}$ can be done considering a traditional NMF algorithm [48] modified to handle the required normalizations. The values of the ground distance parameters $\boldsymbol{d}$ are initialized assigning $d_{qt} = 1$ if $q \neq t$, $d_{qt} = 0$ otherwise.

Step 1. Given $\mathcal{H}_s$ and $\boldsymbol{d}$ the several independent optimization problems associated to distance constraints can be solved finding the optimal values of the flow variables vectors $\mathbf{g}^{ij}$, $\mathbf{g}^{lm}$:

$$\begin{aligned} \mathbf{g}^{ij} &= \arg\min_{\mathbf{g}} \mathcal{D}_{\boldsymbol{d}}(\boldsymbol{h}_i^s, \boldsymbol{h}_j^s) \quad \forall i, j \\ \mathbf{g}^{lm} &= \arg\min_{\mathbf{g}} \mathcal{D}_{\boldsymbol{d}}(\boldsymbol{h}_l^s, \boldsymbol{h}_m^s) \quad \forall l, m \end{aligned} \tag{3.8}$$

where $\boldsymbol{h}_i^s, \boldsymbol{h}_j^s$ are histograms corresponding to the same class while $\boldsymbol{h}_l^s, \boldsymbol{h}_m^s$ are associated to different classes.

Step 2. Given $\boldsymbol{d}$, $\mathbf{W}$ fixed, find $\boldsymbol{p}^k$ and the flow variables $\boldsymbol{f}$. The optimization problem which must be solved is formulated as:

$$\begin{aligned} \min_{p_q^k, f_{q,t}^i \geq 0} \quad & \sum_{i=1}^{N} \sum_{q=1}^{M} \sum_{t=1}^{M} d_{qt} f_{qt}^i \\ \text{s.t.} \quad & \sum_{q=1}^{M} f_{qt}^i = h_i^t, \quad \forall i, \ \forall t \\ & \sum_{t=1}^{M} f_{tq}^i = \sum_{k=1}^{K} w_i^k p_q^k \quad \forall i, \ \forall q \\ & \sum_{q=1}^{M} p_q^k = 1 \ \ \forall k \end{aligned} \tag{3.9}$$

Algorithm 5 EMD-NMF with ground distance learning

Input: $\mathcal{H} = \{\boldsymbol{h}_i\}_{i=1}^{N}$, $\mathcal{H}_s = \{(\boldsymbol{h}_i^s, \boldsymbol{h}_j^s, y_{ij}^s)\}_{i,j=1}^{N_s}$.
  Initialize $\mathbf{W}$ with a traditional NMF algorithm [48].
  Normalize $\mathbf{W}$ such that $\sum_k w_i^k = 1,\ \forall i = 1, \dots N$.
  Initialize $\boldsymbol{d}$ setting $d_{qt} = 1$ if $q \neq t$, $d_{qt} = 0$ otherwise.
  while not converged
    Given $\mathcal{H}_s$ solve the set of transportations problems (3.8) with respect to $\mathbf{g}^{ij}$,$\mathbf{g}^{lm}$.
    Given $\boldsymbol{d}$, $\mathbf{W}$, solve (3.9) with respect to $\boldsymbol{p}^k$, $\boldsymbol{f}$.
    Given $\boldsymbol{d}$, $\mathbf{P}$, solve (3.10) with respect to $\mathbf{W}$, $\boldsymbol{f}$.
    Given $\mathbf{W}$, $\boldsymbol{p}^k$, $\boldsymbol{f}$, $\boldsymbol{g}$ solve (3.11) with respect to $\boldsymbol{d}, \boldsymbol{\xi}$.
  end
Output: $\mathbf{W}$, $\boldsymbol{p}^k\ \forall k$.

This is a simple LP which can be solved efficiently with standard solvers.

Step 3. Given $\boldsymbol{d}$, $\boldsymbol{p}^k$ fixed, find $\mathbf{W}$ and the flow variables $\boldsymbol{f}$. The optimization problem which must be solved is:

$$\begin{aligned} \min_{w_i^k, f_{q,t}^i \geq 0} \quad & \sum_{i=1}^{N}\sum_{q=1}^{M}\sum_{t=1}^{M} d_{qt} f_{qt}^i \\ \text{s.t.} \quad & \sum_{q=1}^{M} f_{qt}^i = h_i^t, \quad \forall i,\ \forall t \\ & \sum_{t=1}^{M} f_{tq}^i = \sum_{k=1}^{K} w_i^k p_q^k \quad \forall i,\ \forall q \\ & \sum_{k=1}^{K} w_i^k = 1, \quad \forall i \end{aligned} \tag{3.10}$$

As in Step 2, this problem is a linear program which we solve using standard solvers.

Step 4. Given $\boldsymbol{p}^k$, $\mathbf{W}$, $\boldsymbol{f}$, $\boldsymbol{g}$ find the ground distance parameters $\boldsymbol{d}$. The optimization problem which must be solved is a quadratic program (QP), *i.e.* :

$$\begin{aligned} \min_{\boldsymbol{d}, \boldsymbol{\xi} \geq 0} \quad & \|\boldsymbol{d}\|^2 + \lambda_1 \sum_{i=1}^{N}\sum_{q=1}^{M}\sum_{t=1}^{M} d_{qt} f_{qt}^i + \lambda_2 \sum_{ijkl} \xi_{ijkl} \\ \text{s.t.} \quad & \text{vec}(\boldsymbol{d})^T (\mathbf{g}^{ij} - \mathbf{g}^{lk}) \geq 1 - \xi_{ijkl} \\ & d_{qt} = d_{tq}, \quad d_{tt} = 0, \quad \forall q, t = 1, \dots, M \end{aligned} \tag{3.11}$$

Algorithm 5 summarizes the main steps of the proposed method.

3.4.1.3 Discovering High Level Activities with Semi-supervised EMD-NMF

In this section we describe how the matrix factorization approach presented in Sec.3.4.1.1 can be applied to the problem of the analysis of dynamic scenes recorded from surveillance cameras.

We first compute level features from the video. Specifically we use a GMM-based background subtraction algorithm [40] to calculate for each pixel the foreground/background information. We also use a KLT tracker [80] to compute the optical flow, which measures the spatial shift between to consecutive frames of selected interest points. If the spatial shift is less then a threshold T_{of} (*e.g.* 2 pixels) the point is considered static and thus discarded.

We divide the scene of interest in $n_x \times n_y$ patches and for each frame we combine foreground with optical flow information. For each patch the median optical flow direction is computed (in order to filter out noise) and it is quantized according to $N = 8$ directions. Patches with a percentage of foreground pixel major then a threshold T_{fg} (*e.g.* 50%) and no optical flow are considered as static. The active patches in a frame corresponds to elementary activities defined by 3 bins length vectors, which identify the position in the scene (x_c,y_c) and the motion direction ('0' for static, '1-9' for moving).

We collect a set of elementary activities over a sequence of frames, long enough in order to guarantee enough variety of events, and we use a standard k-means algorithm to compute a codebook of n_t words. After the codebook is defined, we extract the elementary activities from our sequence and quantize them according to the computed codebook. The temporal sequence is divided into short video clips and for each clip a histogram of occurred events is collected. The final clip histogram $\boldsymbol{h}_i \in I\!R^{n_t}$ is normalized to sum 1. As further described in the experimental section we manually annotate a small set of pairs of clips if they represent similar or different high level activities (*e.g.* vertical or horizontal traffic flows) in order to build the set $\mathcal{H}_s$. From this, a set of N_q quadruple $\{\boldsymbol{h}_i^s, \boldsymbol{h}_j^s, \boldsymbol{h}_m^s, \boldsymbol{h}_l^s\}$ is then selected in order to be fed to the optimization problem (3.11). Note that this set is selected randomly, thus probably generating some redundant information. Then EMD-NMF is used to compute the prototype vectors $\boldsymbol{p}^k$ representing the salient activities and to learn the distance metric $\boldsymbol{d}$.

TABLE 3.4: Datasets and experimental setup.

	n^{o}frames	fps	frame size	n^{o}clips	n^{o}words
Basket	6000	12	368 × 320	100	16
Junction	12000	25	360 × 288	40	16

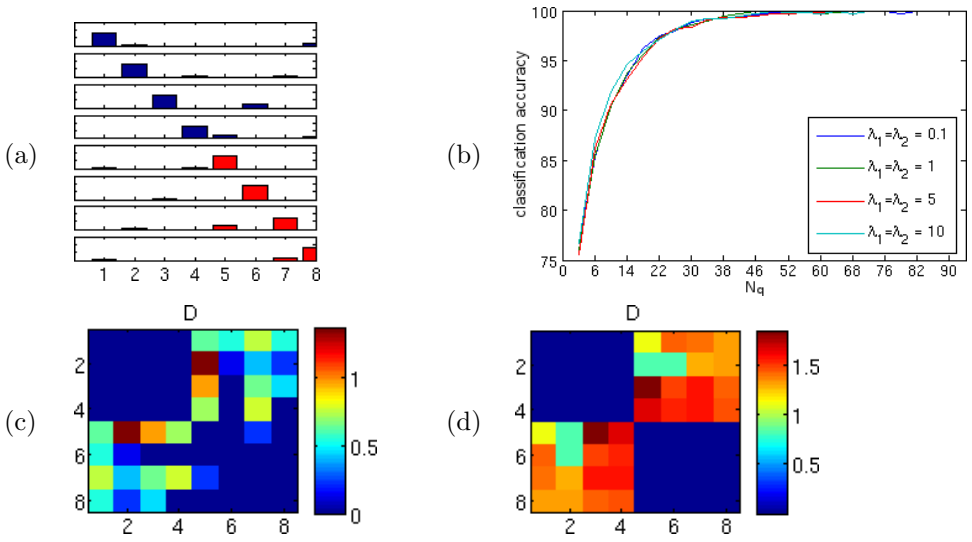


FIGURE 3.11: (a) Synthetic input data, with $m = 8$ and $N = 8$. (b) Performance at varying N_q. Ground distance matrices $\boldsymbol{d}$, learned with (c) $N_q = 30$ and (d) $N_q = 40$. The final accuracy obtained is respectively equal to (c) 87.5% and (d) 100.0%.

3.4.2 Experimental Results

3.4.2.1 Datasets and Experimental Setup

We tested the effectiveness of our approach on two public datasets, QMUL Junction[¶] and APIDIS basket[‖]. The Junction dataset depicts a crowded traffic scene, while APIDIS shows a basketball game. The visual vocabularies used in these experiments are the same as in [1, 3]. More details on the datasets and the vocabulary used are reported in Table 3.4. We also show the performance of our method on synthetic data. In our belief, synthetic data can help giving the reader an intuition of the method's working principles, and to understand the effect of varying the parameters values.

3.4.2.2 Synthetic Data

Consider synthetic input data as in Fig. 3.11(a). The color identifies to which of the two classes, *red* or *blue*, the histogram belongs. The performance on classification at varying

[¶]http://www.eecs.qmul.ac.uk/~jianli/Junction.html
[‖]http://www.apidis.org/Dataset

(a) 

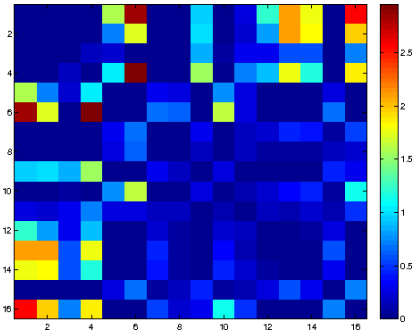
(b) 

FIGURE 3.12: Basket dataset. (a) visual vocabulary and (b) ground distance matrix $\boldsymbol{d}$ learned with $N_q = 1000$.

TABLE 3.5: Performance on APIDIS Basket dataset.

	pLSA	pLSA-bin	EMD-L1 [3]	our method
Basket	94.0%	92.0%	98.0%	98.0%

N_q is shown in Fig. 3.11(b). The final ground distance matrix $\boldsymbol{d}$ obtained for different values of N_q is shown in Fig. 3.11(c,d). It can be easily observed that by increasing the number of quadruples N_q it is possible to get better performance, as well as a more meaningful distance matrix $\boldsymbol{d}$. In Fig. 3.11(d), the learned ground distances between bins 1-4 are zero, meaning that these are highly correlated. The same consideration holds for bins 5-8. The highest ground distance is between bin 3 and 5, which means that these two bins help to discriminate between the two classes *blue* and *red*. Differently from Fig. 3.11(d), in Fig. 3.11(c), the learned ground distance between bins 3,6 and 4,6 is low, which means that the set of quadruples given to learn $\boldsymbol{p}^k$, $\mathbf{W}$ and $\boldsymbol{d}$ is not representative enough.

3.4.2.3 APIDIS Basket Dataset

The results obtained on the basket dataset are reported in Fig. 3.12-3.16. The two events to be discoverd, highlighted in *blue* and *green*, correspond respectively to the events i) '*ball in possession of the yellow team*' and ii) '*ball in possession of the blue team*'. Figure 3.13 shows the prototypes computed based on the groundtruth, *i.e.* obtained by averaging over the clip histograms with the same groundtruth label. In Fig. 3.14 we can observe that the prototypes learned with $N_q = 1000$ are similar to the ones computed based on the groundtruth. However, the learned prototypes $\boldsymbol{p}^k$ contains less active bins, *e.g.* the word 16 is missing from the learned *blue* prototype (Fig. 3.14).

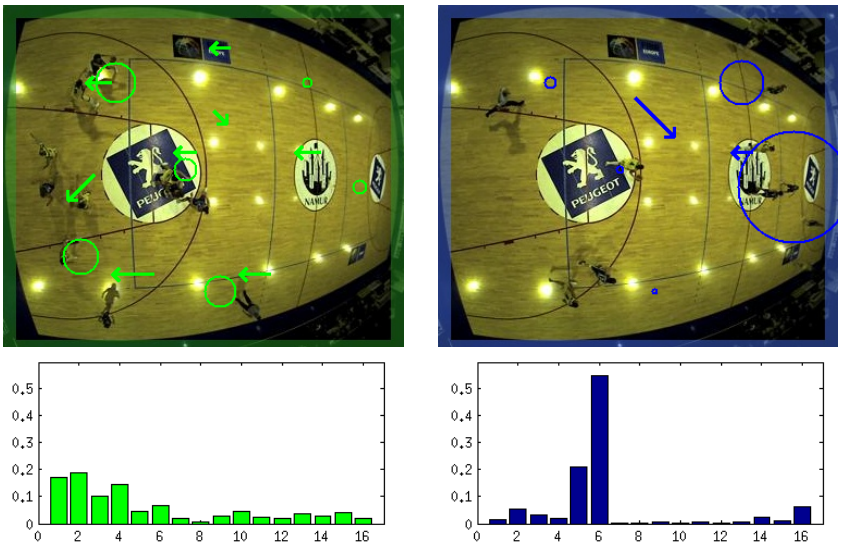


FIGURE 3.13: Basket dataset: prototypes p^k computed based on groundtruth. (Left) ball in possession of blue team (right) ball in possession of yellow team.

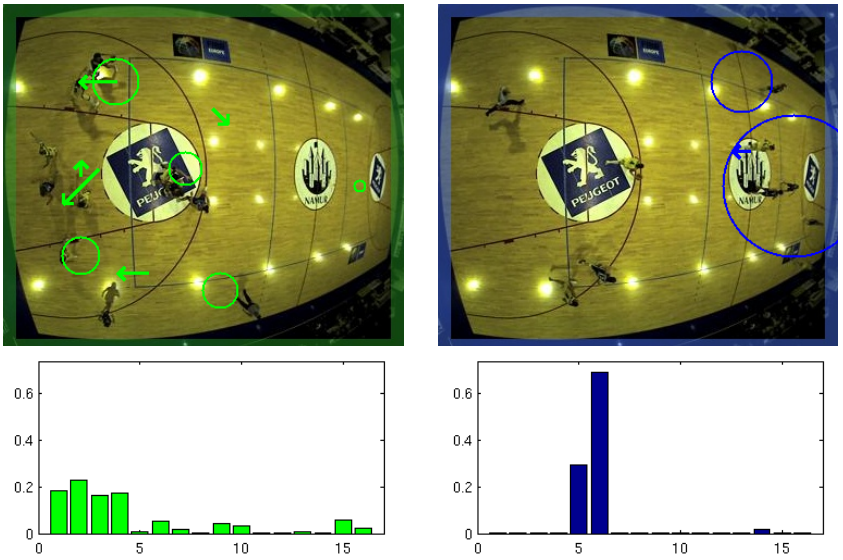


FIGURE 3.14: Basket dataset: prototypes p^k learned with $N_q = 1000$. (Left) ball in possession of blue team (right) ball in possession of yellow team.

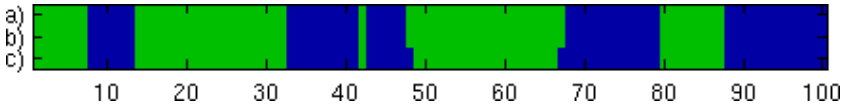


FIGURE 3.15: Basket dataset: temporal segmentation bar obtained with (a) EMD-L1 [3], (b) our method and (c) groundtruth.

This is compensated by the fact that the learned ground distances between the word 16 and the words $5, 6, 14$ is low, which allows to associate clips with high occurrence of word 16 to the *blue* event (Fig. 3.12(b)). In Fig. 3.13 we can see that the *green* event is described mostly by the presence of words $1 - 4$ and 15. However, these words

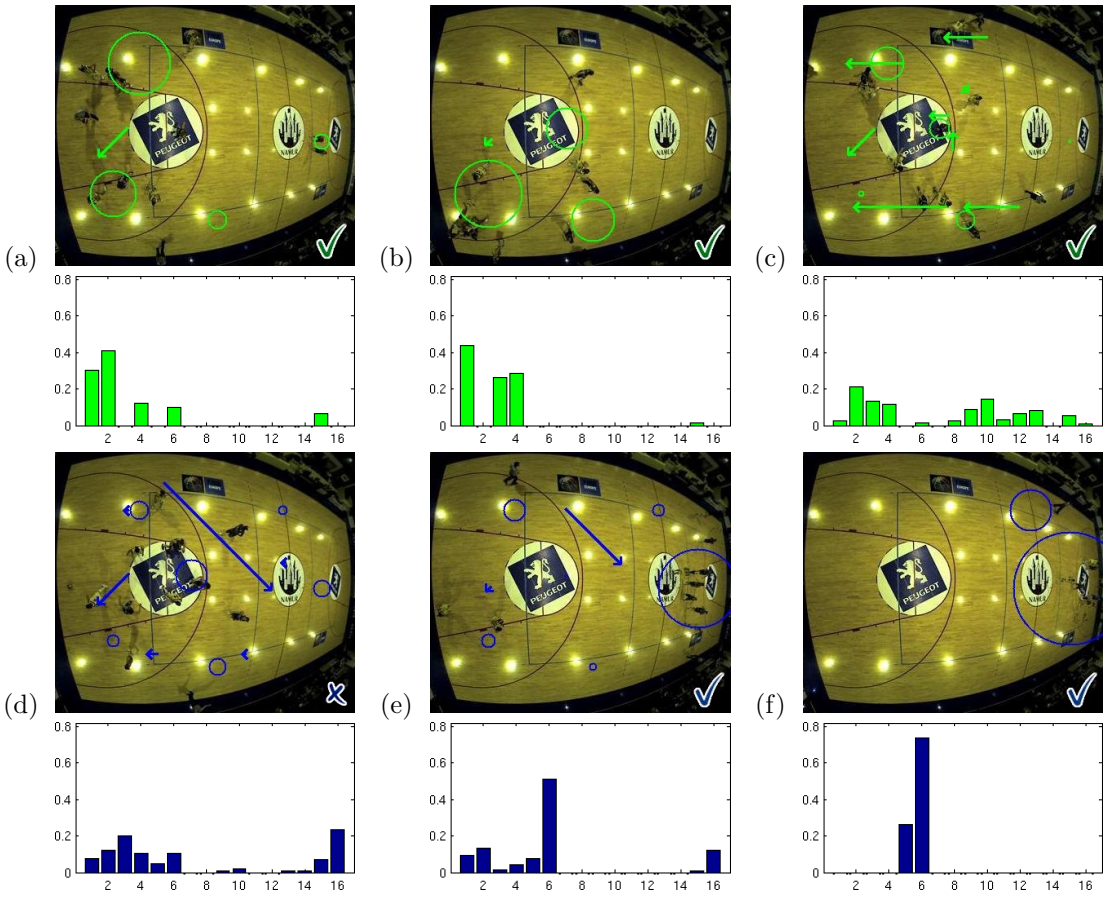
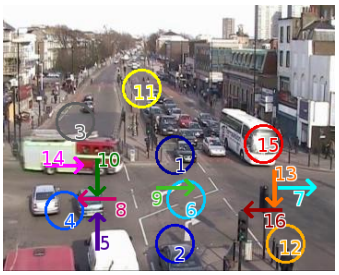


FIGURE 3.16: Basket dataset: sample keyframes with correspondent bag-of-words are shown. The color indicates their ground truth label, respectively (a-c) *green* and (d-f) *blue*. All the shown clips, except (d), are correctly classified by our method. Clip (d) is an example of misclassification, corresponding to clip 68 in the temporal bar of Fig. 3.15. This clip corresponds to the event of '*ball in possession of the yellow team*', but the player configuration is still very similar to the '*ball in possession of the blue team*' event.

may not show up together at the same time (*e.g.* the occurrence of word 2 is zero for clip (a)), while words typical of one event can occur during a different one (*e.g.* occurrence of word 2 is non-zero for clip (d,e)). The ground distance learned in this case help to compensate this noisy effect, intrinsic of the nature of the events (*i.e.* the game patterns played by a team will tend to be similar, but not completely identical), by balancing the words distribution among clips, in other words, it emphasizes the occurrence of group of activities w.r.t. the occurrence of a single activity. Figure 3.15(b) shows the final temporal segmentation obtained with our method, which corresponds to an accuracy of 98%, which is comparable with the results obtained in [1, 3], see Table 3.5 and Fig. 3.15. To be specific, the groundtruth used in [1, 3] was set at frame level, while in this work we converted the groundtruth to clip level, thus all the frames belonging to the same clip has the same label. The changes slightly the result, *i.e.*

TABLE 3.6: Performance on QMUL Junction dataset.

	std pLSA [25]	hrc pLSA [25]	EMD-L1 [3]	our method
Junction	90.0%	77.5%	92.5%	92.5%

(a) 

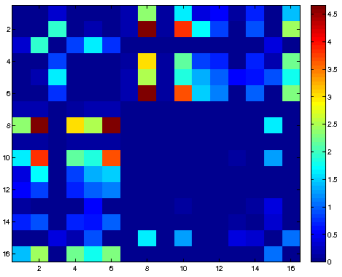
(b) 

FIGURE 3.17: Junction dataset. (a) visual vocabulary and (b) ground distance matrix $\boldsymbol{d}$ learned with $N_q = 1000$.

from 92.25% to 92.0%, and allows an easier verification of the method's accuracy on the classification performance task (see Fig. 3.15 and Tab. 3.5 for comparison). It is important to notice that the two misclassification corresponds to transition clips, which collect words occurrence from both the '*green*' and '*blue*' events. For example, in clip 68, whose bag-of-words representation is shown in Fig. 3.16(d), players of the yellow team have just enter in possession of the ball, but they seem to delay the move to the opposite game court, thus making the game configuration more similar to the '*green*' event, *i.e.* when the blue team is on attack. In this case of 'transition clip', the classification error is due to the mixed nature of the clip, rather than to the failure of our method.

3.4.2.4 QMUL Junction

Similar observations discussed for the Basket dataset can be drawn for the Junction dataset. The results obtained on Junction datasets are reported in Fig. 3.17-3.20. The two events, *blue* and *green* to be discovered within this dataset correspond respectively to i) '*vertical traffic flow*' and ii) '*horizontal traffic flow*', where this last on includes alternate '*from left to right*' and '*from right to left*' horizontal flow. As we can see from Fig. 3.13, the most discriminative words for the '*vertical flow*' event are $1, 2, 5, 6$, while words $8, 14, 16$ are mostly characterizing the event '*horizontal flow*'. In Fig. 3.17(b) we can verify that the learned ground distances among the words inside of each group are low, while among two words of different groups are high. Figure 3.20 shows the obtained video segmentation results, while a quantitative evaluation is reported in Table 3.6.

FIGURE 3.18: Junction dataset. Prototypes p^k computed based on groundtruth: (left) horizontal and (right) vertical traffic flow.

FIGURE 3.19: Junction dataset. Prototypes p^k learned with $N_q = 1000$: (left) horizontal and (right) vertical traffic flow.

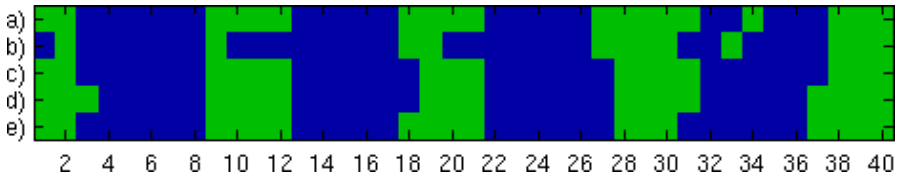


FIGURE 3.20: Junction dataset: temporal segmentation results obtained with (a) standard pLSA [25], (b) hierarchical pLSA [25], (c) EMD-L1 [3], (d) our method and (e) groundtruth. The final accuracy obtained with our method is 92.5%.

3.5 Long-Term Behavioral Pattern Analysis in Complex Urban Scenes

3.5.1 Method

Our method employs two main steps: (i) FG detection based on BG modeling and, (ii) analysis of behavioral patterns, based on the extracted FG information (Sec.3.5.1.3). In order to improve the performance at step (i), we propose a new BG subtraction method where the BG model is built on a sparse representation via a feature dictionary that is learned from the input data (Sec.3.5.1.2). We provide a general overview on Sparse Coding and Auto-Encoders in Sec.3.5.1.1.

3.5.1.1 Learning Local Features Dictionary with Auto-Encoders

Sparse Coding. In sparse signal modeling, input signals are represented as a (often linear) combination of a few coefficients selecting atoms in some over-complete bases or dictionary $D = \{\Phi_j\}$. Formally,

$$x = \sum_{j=1}^{M} a_j \Phi_j + \epsilon \tag{3.12}$$

Here $x \in \mathbb{R}^N$, $\mathbf{a} = \{a_j\} \in \mathbb{R}^M$, $D = \{\Phi_j\} \in \mathbb{R}^{N \times M}$, generally $M > N$, and ϵ is the approximation error. Note that when $\mathbf{a}$ is sparse, most of the bins of $\mathbf{a} \in \mathbb{R}^M$ are zero. Formally, we can write this sparsity condition as $||a_j||_0 = K$, $K < M$. In other words, while x is mapped to a higher dimensional space (*i.e.* from $\mathbb{R}^N$ to $\mathbb{R}^M$, with $M > N$), generally the dimension of its sparse representation is lower than the initial space dimension (*i.e.* $K < N < M$).

It has been shown that mapping the data into a significantly higher dimensional space with an over-complete basis dictionary can lead to superior performance in many applications [76, 84]. In this work, we investigate the effect of using sparse coding for background modeling. Relative to prefixed dictionaries such as wavelets, learned dictionaries bring the advantage of better adapting to the images, thereby enhancing the sparsity [85]. We learn our basis dictionary $D = \{\Phi_j\} \in \mathbb{R}^{N \times M}$ through sparse Auto-Encoders. We briefly review auto-encoders below. For more detailed explanation we refer readers to [72].

Auto-Encoders. Auto-encoders (AE) are unsupervised models that learn a compressed representation for a set of data. Specifically, auto-encoders learn a function $h(\cdot)$ that maps an input vector $x \in \mathbb{R}^N$ to a feature vector $a = h(x) \in \mathbb{R}^M$, together with a function $g(\cdot)$, that maps $h(x)$ back to $\hat{x} = g(h(x)) \in \mathbb{R}^N$, where $\hat{x}$ is the reconstruction of the input vector x. In the AE notation, N and M are respectively the number of *visible* and *hidden* units.

The functions $h(\cdot)$ and $g(\cdot)$, named respectively *encoder* and *decoder* function, are computed in a way that minimizes the reconstruction error between the two vectors x and $\hat{x}$. The Encoder function $h(\cdot)$ is defined as: $a = h_{W,b}(x) = s(W_1 x + b) = s(z)$, where s is the *activation function*. In this case we chose s to be the sigmoid function $s(z) = \frac{1}{1+e^{-z}}$. Estimating $h(\cdot)$ corresponds to the estimation of the parameters $W \in \mathbb{R}^{M \times N}$, which is the *weight matrix*, and $b \in \mathbb{R}^M$, which is the *bias vector*. The Decoder function $g(\cdot)$ is defined as: $g_{W,c}(z) = s(W_2 z + c)$. Given a set of p input vectors $x^{(i)}, i = 1, ..., p$, the weight matrices W_1 and W_2 are adapted using backpropagation to minimize the reconstruction error. The cost function to be minimized is therefore:

$$J(W, b) = \min_{W,b} \sum_{i=1}^{p} ||x^{(i)} - \hat{x}^{(i)}||_2 \tag{3.13}$$

where $\hat{x}^{(i)}$ is dependent implicitly on $\{W, b\}$ and $|| \cdot ||_2$ is the Euclidean distance. This step can be performed via batch gradient descent or more sophisticated algorithms like conjugate gradient or L-BFGS to speed up the performance [72]. A penalty term is also added in the optimization function to force the learned features to have desirable properties. There are many sophisticated versions of auto-encoders; differences arise essentially in the specifics of the assigned penalty term. In our case, we use sparse auto-encoders, which force the average hidden unit activation to be sparse [86]. This is done by designing a *penalty term*, which enforces the activation of a hidden unit $\hat{\rho}_j = \frac{1}{p} \sum_{i=1}^{p} [a_j(x^{(i)})]$, $j = 1, \cdots, M$ to be close to a desired value, $\hat{\rho}_j = \rho$, where ρ is the *sparsity parameter*. The overall cost function is now:

$$J_{sparse}(W, b) = J(W, b) + \beta \sum_{j=1}^{M} KL(\rho || \hat{\rho}_j) \tag{3.14}$$

where the weight parameter β controls the relative importance of the *sparsity penalty* term, and KL is the Kullback Leibler distance.

A *weight decay* term on W is also added, whose importance is regulated by the parameter λ, in order to penalize the magnitude of the weight and prevent overfitting. The parameters used in this framework are therefore: (i) the number of *hidden units* M; (ii) the *sparsity* parameter ρ; (iii) the *weight* of the *sparsity penalty* term β; (iv) the *weight decay* parameter λ.

Learning Feature Dictionary for Sparse Representation of Local Patches. In the context of generic image understanding, instead of the whole image sparse coding is applied to local parts or descriptors [84]. In these approaches, the input signal x usually corresponds to a small image patch of $\sqrt{N} \times \sqrt{N}$ pixels which is stacked as a vector $x \in \mathbb{R}^N$. Similarly our approach first randomly samples a set of p patches $\{x_i\}$ from a sufficiently representative training sequence of images. Auto-encoders with parameters $\theta = [W, b]$ are then learned using these samples. The feature dictionary discovered thus captures the most basic and typical visual patterns presented in the training images set.

3.5.1.2 Background Subtraction

We build upon [61] for background modeling. As a novelty, we propose to include more discriminative auto-encoder features, besides the *rgb* values, in the background model. For each pixel *(i,j)* we consider the sparse representation $a \in \mathbb{R}^M$, obtained by mapping the patch centered at *(i,j)* via the *Encoder* function $h(\cdot)$ (previously learned, as explained in Section 3.5.1.1). Note that general sparse methods for background subtraction consider the image as a whole [62, 64–66] and no attempt has been made to date to build a sparse vocabulary of local patches. The background model for each pixel is then built as in [61], modeling the pixel features distribution as a Mixture of K Gaussians:

$$p_i(x) = \sum_{k=1}^{K} \pi_k e^{-\frac{1}{2}(x-\mu_k)'\Sigma_k^{-1}(x-\mu_k)} \tag{3.15}$$

where $\Sigma \in \mathbb{R}^{D \times D}$ and $D = 3 + M$. For computational efficiency, an assumption of dimensionality independence is made. In this way, the full covariance matrix simplifies to a diagonal matrix: $diag(\Sigma) = [\sigma_1^2, ..., \sigma_D^2]$. A second assumption that the variance is the same in each direction (*i.e.* $\sigma_1 = \sigma_2 = ... = \sigma_D$) simplifies Σ as follows: $\Sigma = \sigma^2 I$, where

$I \in \mathbb{R}^{D \times D}$ is the identity matrix. This simplifies formula 3.15 as follows:

$$p_i(x) = \sum_{k=1}^{K} \pi_k e^{-\frac{(x-\mu_k)'(x-\mu_k)}{2\sigma_k^2}}$$

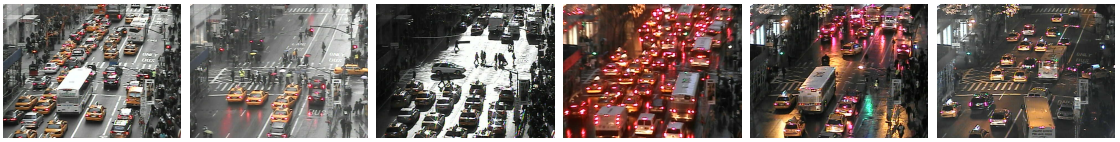
While these assumptions have proven to be effective for a mixture of Gaussians based on *rgb* features, a priori, we cannot expect that methods based on them will work with different types of features. We will see in Section 3.5.2 that these assumptions also hold in our case. The parameters involved in this framework of adaptive mixture model for BG subtraction are the learning rate α and the threshold T_b, that decides whether a point data is well described by the BG model [61].

3.5.1.3 Extracting Typical and Anomalous Patterns of Behavior

Previous works [3, 18, 45] showed that statistical analyses based on simple low-level cues, *e.g.* optical flow, can reveal high-level recurrent patterns of behaviors. However, these analyses are performed on short-term periods, *e.g.* several hours of video. Thus, the recurrent behaviors extracted correspond to such examples as different traffic flows regulated by traffic lights. In this work, we wish to extract significant patterns correlated to human behaviors which are exhibited via long-term analyses; we show how such analyses can be done via a simple measure such as the percentage of foreground pixels, here denoted by τ. The intuition here is that the high-level information τ can be interpreted as an intensity measure of activities happening in a region of interest. For example, when few vehicles are circulating, *e.g.* in the early morning, τ will be low, while during rush hours the measured τ will be higher. An anomalous behavior can be determined by considering the *agreement* on τ of observations taken at a specific day of the week and at a specific time of the day. In details, given N observations τ_i, *with* $i = 1, \ldots, N$, we define the anomaly score as the variance-scaled distance from τ_i to μ_τ:

$$S_i = \frac{|\tau_i - \mu_\tau|}{\sigma_\tau} \tag{3.16}$$

Intuitively, given N observations taken *e.g.* on Monday $9am$, the anomaly score for τ_1 will be higher, w.r.t. $S_2, \cdots, S_N$, if its agreement on μ_τ is lower, w.r.t. agreement given by $\tau_2, \cdots \tau_N$, and so on. Experimental results are in line with our assumptions and they will be discussed in Sec. 3.5.2.



2011-12-10 Sat,15:14 | *2011-12-06 Tue,13:59* | *2011-12-10 Sat,13:24* | *2011-12-07 Mon,16:20* | *2011-12-06 Thu,16:30* | *2011-12-05 Mon,19:18*

FIGURE 3.21: NYC-5th dataset: representative frames of a *complex* and *crowded* scene, challenging for a foreground detection task based on background modeling.

3.5.2 Experimental Results

In this section, we present the dataset and the results obtained with our method.

3.5.2.1 Dataset

Our dataset consists of imagery collected from a public webcam viewing over Fifth Avenue in New York City (NYC-5th)[**]. The data was collected over nearly four weeks, from December 1^{st} to 25^{th}, 2011, at a rate of 2 frames per minute. Frame size is 480 $\times$ 640 pixels. The collected $\sim 72K$ frames require a $12Gb$ storage occupancy. Acquiring the same temporal period at a rate of 1 *fps* would have required 225 *Gb* storage. Sample images of this dataset are shown in figures 3.21. In order to evaluate the accuracy of FG detection, on which we rely for further analysis, a sequence of frames has been annotated with the ground-truth FG mask. In particular, two frames per hour, on Dec.6^{th}, have been annotated (*i.e.* we picked frames at times 00:00, 00:30, 01:00, and so on). We selected this day as it contains a large variety of light and weather changes. Data and ground truth are available online on the author's website[††]. Our hope is that this dataset can contribute as a reference benchmark for long-term activity analysis, as well as for background subtraction (BS) methods evaluation on complex and crowded sequences. Available benchmarks for BS evaluation, generally consist of short sequences, with a few annotated frames [87], or consist of artificially generated sequences [59].

3.5.2.2 Learning a Vocabulary for Sparse Patch Representation

As a first step, we train the auto-encoders in order to generate the features that allow a sparse representation of the video data. We set the dimension of the patch to

[**]http://www.earthcam.com/usa/newyork/fifthave
[††]http://disi.unitn.it/∼ zen

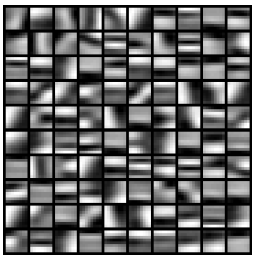
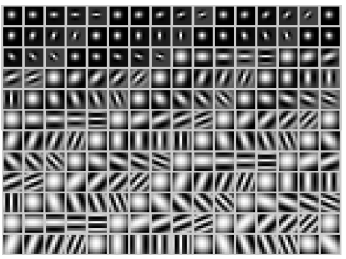
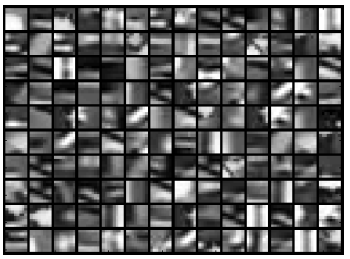
a)  b)  c) 

FIGURE 3.22: (a) Features learned with Auto-Encoders on NYC-5th dataset, (b) examples of Gabor Filters and (c) set of patches randomly sampled from NYC-5th dataset.

$\sqrt{N} = 8$. This value is a good compromise as it allows us to learn sufficiently discriminative features; larger patches would have a higher probability of including foreground. Figure 3.22(a) shows the set of features learned by randomly sampling in space and time $p = 20000$ patches from a 1-day-long sequence (*i.e.* Dec.6^{th}) and using the following setting: $M = 128$, $\rho = 0.003$, $\beta = 3$, and $\lambda = 0.0001$. Training the auto-encoders with these settings on an Intel(R) Core(TM) i5 CPU, 2.67GHz requires about 20 minutes. Figure 3.22(a) can help with understanding the meaning of the weight matrix W and the hidden representation a. Each column of the weight matrix $W \in \mathbb{R}^{N \times M}$ is reshaped to form a $\sqrt{N} \times \sqrt{N}$ patch. The $M = 128$ filters obtained are displayed. When a patch x is mapped via $h(\cdot)$ to a $\mathbf{a} = \{a_j\} \in \mathbb{R}^M$ sparse representation, the non-zero coefficients of $\mathbf{a}$ identify the features that better represent the signal x. It is well known that features learned via auto-encoders at one layer resemble Gabor-like filters. However, it is also known that learning features from data often achieves a sparser representation as the learned dictionary better fits the data [85]. This can be practically confirmed by observing the learned features and some Gabor filters, shown respectively in Fig. 3.22(a) and (b), while in Fig. 3.22(c) some sample patches randomly extracted from the original sequence are shown. It is quite straightforward to see that the filters learned with AE highly resemble the original patches. Furthermost, looking at Fig. 3.22(a), we can see that filters with a certain diagonal orientation (from top left to bottom right) are nearly absent because the scene perspective is one where a majority of edges are oriented on the other diagonal direction. Additionally, besides regularly oriented edge gradients, some data-specific shapes can be learned. *Learning* w.r.t. *using hand-designed* features may be a promising approach in the context of visual surveillance and in crowded scenarios like the one we considered, in which: i) both BG and FG signal present a high data redundancy, where the background is constant and similar elements tend to appear repetitively in space and time, and ii) the variability of features data is limited to the *small world* variability defined by the scene observed through the camera.

3.5.2.3 Foreground Detection

The performance evaluation of the different methods for FG detection is reported in Fig. 3.23 and Tab. 3.7. The performance is measured in term of *precision* and *recall* which quantify, respectively, the number of pixels correctly identified as FG, divided by the number of pixels classified as FG (p) and by the number of pixels defined as FG in the ground truth (r). The F-measure is defined as: $F = (1+\beta^2)\cdot\frac{\text{precision}\cdot\text{recall}}{(\beta^2\cdot\text{precision})+\text{recall}}$, with $\beta = 1$. Fig. 3.23(a) shows the performance of the three methods at varying parameters T_b and α (in details, at varying $T_b^{rgb}, T_b^{ae} \in [3.5, 4.5]$, $T_b^{rgb-ae} \in [7.0, 8.0]$, and $\alpha \in [0.01, 0.10]$). Also, the average error on the FG percentage estimation, ϵ_τ, has been measured. It was observed that the best estimate of τ is obtained with the highest balance between precision and recall (this optimal results area is highlighted in red in Fig. 3.23(a)). Among all possible (T_b, α) values combinations, the one allowing the best performance for each method is shown in Tab. 3.7.

In general, it can be seen that using only *ae* features allows a better performance than using only *rgb*, while the highest accuracy is achieved by combining *ae* and *rgb* features. Moreover, a higher robustness w.r.t. parameters variation is achieved with *rgb-ae*. We observe that our method performs well w.r.t. sudden illumination changes even at a low learning rate, *e.g.* $\alpha = 0.03$, while the method based on only *rgb* performs more poorly at that rate (best accuracy is obtained with $\alpha = 0.10$). Using a high learning rate makes the method less robust to detecting temporary stationary objects (*e.g.* cars stopped at red traffic light). A visual comparison of the two methods performance and

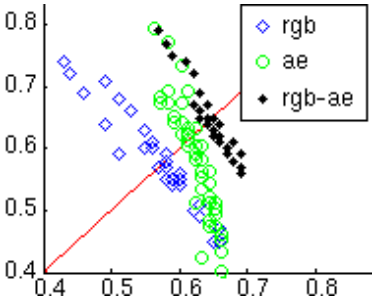


FIGURE 3.23: Overall performance (recall vs. precision)

TABLE 3.7: Best overall performance on foreground detection.

Dec 6th	D	T_b	α_T	precision	recall	F-Measure	ϵ_{FG}
MoG, *rgb* [61]	3	4.0	0.10	0.60	0.56	0.58	4.21 %
MoG, *ae*	128	4.0	0.10	0.62	0.58	0.60	2.89 %
MoG, *rgb-ae*	3+128	7.0	0.03	**0.66**	**0.66**	**0.66**	**2.49** %

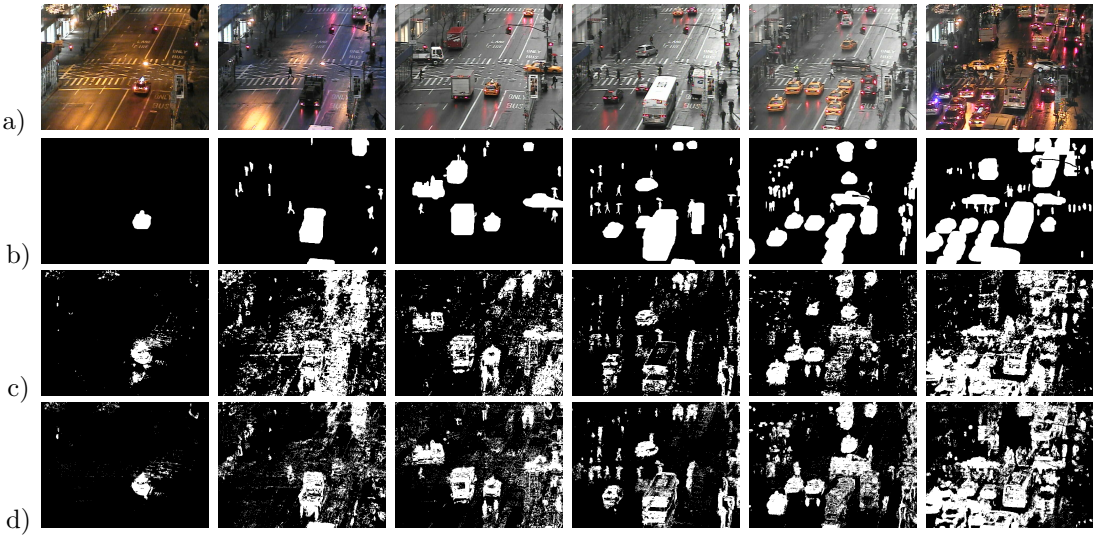


FIGURE 3.24: Performance on foreground detection. (a) original frame (b) ground truth (c) [61] and (d) our method.

the ground truth mask can be observed in Fig. 3.24. Figure 3.25 shows the performance of our method w.r.t. different challenging situations. A video sequence showing the performance of our method during challenging conditions (*e.g.* rain, night) is shown online*.

More experiments have been conducted to explore our method performance at varying AE parameters, *i.e.* number of hidden layer M and sparsity coefficient ρ. A higher performance w.r.t using only *rgb* features has been obtained with $M = 32, 64, 128$, although with $M = 128$, a higher stability has been observed w.r.t. the variation of α and T_b. The best performance is obtained with $\rho \in \{0.02, 0.04\}$, while a drop in accuracy is observed with $\rho > 0.5$. We believe that this finding is evidence of the beneficial effects of using sparse representations. Experiments have been conducted also using HOG and Gabor Filters as local features. However, the measured performance was not satisfactory. Additionally, Gabor filters requires some set up efforts, in order to select the most representative filters for the dataset (Fig. 3.22(a)). We conjecture that the gap in performance is based on the following reason: AE method generates an over-complete dictionary, with bases having very similar but slightly shifted structures (see Fig.2(b)). This results in a sparse representation where the signal is described in terms of indices of active bins. Conversely, features like HOG or Gabor filters map patches to a dense representations, where the signal is characterized in terms of different intensities of bin values. We believe this is a key differentiation that leads to more robust representation for AE methods w.r.t. thresholding operation performed when discriminating BG from FG in our framework. Additionally, we believe that HOG features extracted at 8x8 patch

FIGURE 3.25: Performance on foreground detection w.r.t. different challenges: (a) sudden light changes, (b) sudden image blurring (camera defocus), (c) night lightening with rain, (d) stationary foreground. (Left) original frame, (center) [61], (right) our method.

level may not be able to correctly model the data variability. Our above hypothesis is in line with recent findings on sparse coding [17]. These findings suggest that while hand-designed features like SIFT or HOG perform well on the tasks for which they were initially designed, they often perform poorly on novel scenarios.

3.5.2.4 Extracting Typical and Anomalous Patterns of Behavior

The average traffic intensity measured for the whole dataset with our method is reported in Fig. 3.26. As the traffic intensity variability between successive frames is high, the values are smoothed in time with a median filter of 120 frames length. The behavioral

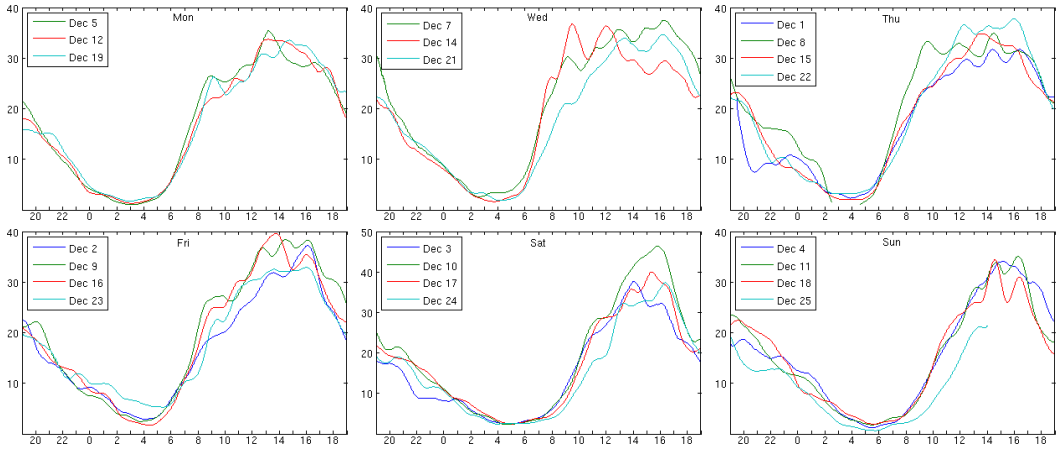


FIGURE 3.26: Patterns of behavior (average traffic intensity per time of the day) measured for the NYC-5th dataset.

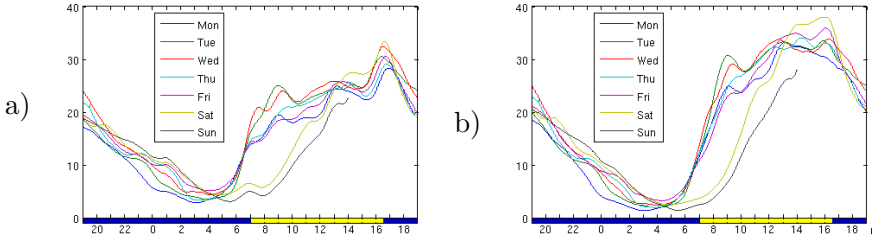


FIGURE 3.27: Typical patterns of behavior obtained for each day of the week, with (a) [61], based on *rgb*, and (b) our method, based on *rgb-ae* features.

pattern of Tuesday was found to be similar to the pattern of Wednesday and it was omitted for space reasons. In some cases, *e.g.* Thursday 8^{th} $4am$, the graph is interrupted because of missing data. In the next two sections, we discuss the results on typical patterns and anomalous behaviors extracted.

Typical patterns of behavior. The average typical behaviors per day of the week, obtained with [61] and with our method are shown respectively in Fig. 3.27(a) and Fig. 3.27(b). As highlighted in Fig. 3.27(b), peaks in patterns due to sudden light changes, e.g. around $7am$ (sunrise) and $4{:}30pm$ (sunset), are reduced significantly. Per the findings highlighted in Fig. 3.27(b), we observe the following: (i) Daily traffic intensity patterns are very similar to each other, *i.e.* the average traffic intensity does not vary much, given a day of the week and time of day. (ii) Two main daily behavioral patterns can be observed, one for the working days (*Mon-Fri*) and one for the weekend (*Sat,Sun*). For the second one, the morning rise in activity tends to start much later. (iii) At night, we observe an incremental drop in traffic intensity, and the average traffic intensity is sorted w.r.t. to the day of the week, going from the lowest on Sunday night,

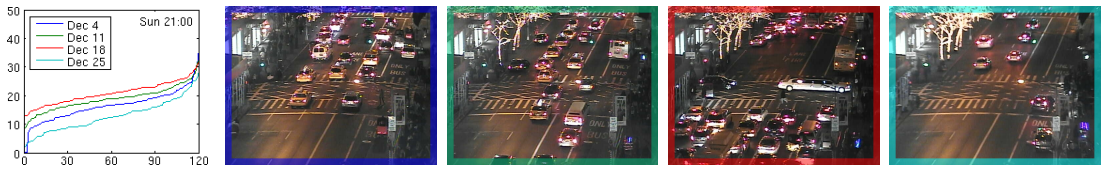


FIGURE 3.28: Answering queries like: *How does a typical Saturday evening in New York look?* It can be easily observed that the lowest traffic density was recorded on Christmas Eve.

FIGURE 3.29: Detecting anomalous behaviors: (a) *Dec. 1^{st}, Thu, 9:00 pm*, regular traffic flow is limited due to a pedestrian demonstration. (b) *Dec. 25^{th}, Sun, 9:45 am* unusual lower traffic intensity due to being Dec. 25^{th} the day of Christmas. (c) *Dec. 8^{th}, Thu, 9:30 am*, anomalous peak in traffic intensity.

to the highest on Saturday night. Indeed, even in New York City, *the city that never sleeps*, people seem to have more bedtime before the beginning of new work weeks.

An example of a *typical Saturday night* at 9*pm* is shown in Fig. 3.28. For each Saturday in our NYC 5th Avenue dataset, the *median frame* (i.e. the frame associated to the median τ value) within the time interval from 8:30 to 9:30*pm* is automatically extracted. We plot in Fig. 3.28(left), the traffic intensity measured for each of the 120 frames, sorted from the lowest to the highest value. While Dec. 4^{th}, 11^{th} and 18^{th} look very similar, on Christmas Eve (Dec. 25^{th}), a lower traffic intensity is observed.

Anomalous activities. Our method can be employed for the automated detection of anomalous behavior w.r.t. the typical patterns learned. Figure 3.29 depicts three main anomalies detected with the method: (a) *Thu, 9pm*, unusual low traffic intensity due to the occurrence of a pedestrian demonstration on a Wednesday night and (b) *Sun, 9:45am*, unusual low traffic intensity recorded on the day of Christmas in the morning,

(c) *Thu, 9:30am*, unusual high traffic intensity. Our intuition is that this traffic peak is due to a traffic congestion at rush hour. In general, it is not always straightforward to identify the specific reason for an atypical density, but such anomalous situations can frame the search for potential explanation from the many events that occur in cities and their influences on the region at the focus of attention. In Fig. 3.29(a) we can note that, while the behaviors on Dec. 8, 15 and 22 around $9pm$ look surprisingly similar among each others (see blue lines in the graph), the behavior on Dec.1 (see red line) differs from them remarkably. The same can be observed for Fig. 3.29(b) and (c). A short video with the detected anomalies is available online on the author's website[‡‡].

3.6 Conclusions

We presented three novel approaches for the analysis of high level activities in complex noisy video scenes, which are particularly well suited for three specific case scenarios.

In case of large size visual vocabulary, we proposed a method which combines EMD matrix factorization and sparsity constraints, thus being robust to features' noise and producing as output a set of sparse bases. This is greatly beneficial for complex scene analysis applications, where multiple activities simultaneously occur in the scene and it is of paramount importance to be able to extract the most relevant elementary activities for automatically inferring high level behaviors. The proposed approach has been used to find recurrent activities in publicly available video datasets and has been extensively compared with state-of-the-art methods. The application of the proposed matrix factorization algorithm is not limited to video data. Indeed, we believe it will be suitable for many other problems, such as data analysis or human behavior understanding.

The use of EMD is highly beneficial because is allows to consider the similarity among atomic activities, which is encoded in the definition of the ground distances. Still, the definition of these distances is crucial for the effectiveness of the proposed approach. We presented EMD-NMF, a semi-supervised method which applied to dynamic scene analysis allows, not only to discriminate among events, but also to learn the relationship of the atomic activities which characterize the event. Differently from dimensionality reduction approaches which map the features vectors to a different space, EMD-NMF

[‡‡]http://disi.unitn.it/ zen/video/artemis13-zen.avi

allows a more intuitive interpretation of the learned relationship. Besides, the required annotations for this semi-supervised task correspond to a set of short length video clips pairs, labeled with *must link* or *cannot link* constraints. While manual annotation is usually a tedious and time consuming process, this kind of required annotation is especially convenient in the considered scenarios (*e.g.* traffic monitoring), where the type of occurring behaviors is not defined a priori. In other words, labeling pairs of short video clips as belonging or not to the same high level event can be way easier than having to assign a label to an event.

In the case of long-term behavioral patterns analysis, we have shown how high-level patterns of behaviors can be extracted from the analysis of low level cues, *i.e.* foreground, to provide insights on the intensity of activities occurring in a city. Additionally, we showed that sparse coding applied at a *patch*- rather than *frame*-level can significantly increase the performance of foreground detection in crowded and complex urban scenarios, thus reducing the noise extracted from the imagery data. Our work is motivated by the pursuit of robust features for a stable background representation in crowded and complex scenes, and the exploration of advantages in using a sparse representation for visual surveillance.

Chapter 4

Personalizing Models for Facial Expression Analysis with Transductive Parameter Transfer

4.1 Intuition

Nowadays, the importance of adaptive and personalized human-computer interfaces, as opposite to systems designed for an "average" user, is widely recognized in a large variety of applications. Machine learning algorithms for automatic analysis of facial expressions and body movements are currently employed in many HCI systems. However, surprisingly, few of these systems adapt the learned models to specific users. The issue with personalization is that typically a significant amount of labeled data is required to train user-specific classifiers. This is practically infeasible in many real world applications as collecting a large number of annotated samples is very time consuming.

In this work we propose a novel transfer learning framework to build personalized models without resorting to user-specific labeled data (Fig. 4.1). Our approach relies on learning a regression function which captures the relation between a data distribution and the classifier learned on the samples generated from that. Specifically, our method is based on three phases (Fig. 4.2). In the first phase, given N auxiliary *source* users and the associated *labeled* training samples, we learn a set of N classifiers, parametrized by the vectors $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N$. In the second phase a regression function $f(\cdot)$, which relates

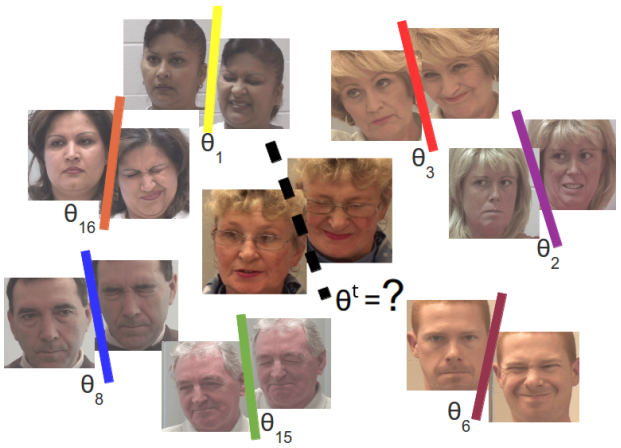


FIGURE 4.1: Human expressions like pain can be exhibited in many different ways. Our goal is to obtain a personalized classifier $\boldsymbol{\theta}^t$ for a new user without acquiring labeled data. We show how $\boldsymbol{\theta}^t$ can be *accurately* and *efficiently* inferred exploiting the similarity between the data distribution of the target user and the distributions from other subjects with known $\boldsymbol{\theta}_i$. The intuition is that, despite the inter-subject variability, knowledge can be transferred among individuals showing similar behavioral patterns.

the *unlabeled* data distribution of the i-th source user with the associated classifier $\boldsymbol{\theta}_i$, is learned. Importantly, once $f(\cdot)$ is obtained, labeled data are not required anymore. Finally, given a novel *target* user, it suffices to apply $f(\cdot)$ to the associated data distribution to obtain the personalized classifier $\boldsymbol{\theta}^t$.

The proposed transfer learning approach is a general framework that can be applied to different types of data, *e.g.* images, audio, text, physiological signals, inertial measurements, etc. In this work we focus on two applications: facial expression analysis from visual data, as the face is one of the main channels through which people convey their emotions [88], and gesture recognition from accelerometer measurements, due to the widespread diffusion of mobile devices and consumer products which integrate inertial sensors (*e.g.* the Nintendo Wii). This second application shows well that the proposed method is quite general and can be applied also to non-visual data, such as the accelerometer-based ones considered.

Specifically, our experimental evaluation is conducted on three different datasets. In a first serie of experiments we consider two publicly available datasets for facial expression recognition. Specifically we use the recent PAINFUL dataset [89], which collects videos of patients with shoulder injuries, and we devise a patch-based facial expression recognition approach based on Local Binary Pattern Histograms (LBPH) [90]. Again, user-specific models are of utmost importance in this context as the way in which the patients spontaneously show pain varies considerably between different people (see

Fig. 4.1). Then, we consider the popular extended Cohn Kanade (CK+) dataset [91] and we learn a set of Action Unit detectors using facial landmarks and SIFT descriptors. In a second serie of experiments we use accelerometers data recorded from a wrist-worn smartwatch to learn personalized gesture recognition classifiers. As discussed in previous works [92, 93], personalization is crucial in this scenario since there is a large inter-subject variability in the way gestures are executed.

In Sec. 4.4 we show that our approach outperforms user-independent classifiers and state of the art personalization methods in the three scenarios, even if the tasks and the adopted features are very different. Moreover, at training time, our algorithm is significantly faster than other domain adaptation techniques, most of which are based on time consuming instance re-weighting strategies. We believe that computational cost is a critical factor for personalization in HCI systems, as user adaptation typically needs to be accomplished in a limited time frame.

As far as we know, this is the first transfer learning approach which proposes to learn a mapping between a data distribution and the corresponding classifier's decision boundary. According to [21], current unsupervised domain adaptation works can be differentiated into instance transfer and feature transfer methods. Conversely, the proposed method aims to directly transfer the parameters of the classifiers from the source to the target domain.

4.2 Related Work

In this section we briefly review the literature related to transfer learning techniques, still image-based facial expression analysis and smartwatch-based gesture recognition.

In the last few years several transfer learning methods have recently become popular in the multimedia and the computer vision fields [94–96] to solve or alleviate the so-called dataset bias problem. Transfer learning aims to improve the learning performance in a target domain using knowledge extracted from related source domains. In [21] a survey on different approaches is presented. According to the type of information transferred from source to target domains, the methods are categorized into *parameter transfer*, *feature transfer* and *instance transfer* approaches.

Parameter transfer methods aim to find a set of parameters or priors shared between the source and the target models. In [97] Yang *et al.* extended standard Support Vector Machines (SVMs) and proposed Adaptive-SVMs. Adaptive-SVMs employ a regularization term to impose the target classifier to be similar to the source one. However, these methods usually require annotated target data, which are typically not easily obtained in HCI scenarios.

Feature transfer methods operate by looking for a shared feature representation for source and target data. For instance, in [98] the input feature vector is augmented by obtaining a novel descriptor composed of a shared, a source-specific and a target-specific part. Similarly, in [99] a shared representation for source and target data in terms of visual attributes is proposed.

Instance transfer approaches [100, 101] are commonly adopted when the target data are unlabeled. For instance, in [100] Gretton *et al.* proposed to compute the centroids of the source and the target distributions and to estimate those source sample weights which reduce the inter-centroid distance in a Reproducing Kernel Hilbert Space. These weights are then used to assign importance to the source samples when training a model for classification on target data. The drawback with most instance transfer approaches [100, 101] is that computing the distance between centroids may poorly approximate the real discrepancy between distributions. We overcome this issue by adopting more accurate approaches to quantify the difference between source and target distributions which are based on specific *kernels for distributions*. Moreover, most instance transfer methods rely on a computationally intense training phase, while our method is very efficient.

In the last few years, research on facial expression analysis has made significant progress. Many approaches have proved to be effective for recognizing simple facial expressions (*e.g.* happiness, sadness, anger, etc.) or alternatively for detecting Action Units (AUs) [88, 102]. State of the art (static) face analysis systems follow a common three steps protocol: face registration, feature extraction (and possibly dimensionality reduction) and classification. Face registration typically relies on localizing anatomically salient facial points [103]. Subsequently, different features are extracted either from the whole face image or from patches centered around (a subset of) the facial landmarks. Finally, the classification step is typically based on algorithms such as SVM, Boosting, Random

Forests [88, 102, 104]. Most state of the art systems are trained and tested in laboratory conditions, with datasets mainly consisting of frontal face images and posed emotions [88, 102, 105]. Very little attention has been paid to realistic scenarios and personalized systems. Many works [106–108] have recently focused on recognizing spontaneous facial expressions and non-basic emotions, *e.g.* pain. However, they are based on user-independent models, *i.e.* on detectors trained on a generic dataset, which aim to be sufficiently representative of many possible sources of variability (*e.g.* illumination conditions, target appearance, etc.). Unfortunately, having at disposal only datasets with few hundreds or thousands of images, generalization to arbitrary conditions is hard to achieve.

To cope with this issue, few works have proposed solutions to integrate weakly labeled or unlabeled data. In [109] Sikka *et al.* adopted a Multiple Instance Learning approach for training a pain expression classifier using video-level labels where frame-level labels are not available. The problem of pain detection is also addressed in [110] where an extension of AdaBoost for user-personalization is proposed both in a supervised and in an unsupervised setting. However, in the unsupervised case, the proposed method did not achieve significant improvement in terms of accuracy with respect to the user independent classifier. In [101] Selective Transfer Machine (STM) is proposed for person-specific AU detection. STM is based on the Kernel Mean Matching algorithm [100], which is modified using an iterative minimization procedure where labeled source data drive a progressive movement of the generic SVM hyperplane toward the target space. Even if effective, this approach is very slow at training time, as the underlying optimization strategy is very time consuming. On the other hand, user-specific adaptation algorithms are required to be computationally efficient to be used in HCI applications. Our method is mainly motivated by this need and our experiments confirm that it is much faster than [101], being its accuracy comparable and even better (see Sec. 4.4).

Automatic gesture recognition tools [111] are key technologies in HCI systems. For many years the research in this topic has been focused on vision-based approaches. More recently the advent of low cost depth sensors and the great diffusion of mobile devices with built-in inertial sensors (*e.g.* smartphones, videogame controllers) has lead to new opportunities. In particular accelerometer-based gesture recognition approaches have proved to be advantageous over traditional visual-based methods, being robust to environmental disturbances and to user movements. Among the several mobile devices

with built-in accelerometers, smartwatches, being low cost and non-obtrusive platforms, are suitable to design gesture recognition interfaces.

Since smartwatches are a relatively new technology, only few previous works exist on the topic. Of particular relevance is the work in [112], which specifically addresses the use of smartwatches as gesture-based input devices, underlining the fundamental distinction between the two tasks of gesture recognition and activity recognition. Accelerometer-based gesture recognition is generally modeled as a classification problem, with different works proposing different machine learning algorithms. In particular, SVMs [113, 114], Hidden Markov Models (HMMs) [115, 116] and Bayesian Networks [117] have proved to be effective for this task. Methods based on generative approaches like HMMs are generally limited in their applicability due to their computational complexity. SVMs on the other side usually offer lower computational requirements at classification time, making them preferable for real-time applications on low power devices. Recently, some works highlighted the importance of devising user-specific classification models. Liu *et al.* [93] proposed an approach based on Dynamic Time Warping, one-shot learning and continuous update though template adaptation. Similarly, Mantyjarvi *et al.* [116] presented a system which can be trained with a single gesture and employs noise-distorted copies of that gesture to learn a HMM. Costante *et al.* [92] proposed a metric learning algorithm for building personalized classifiers. However, all these approaches rely on labelled user-specific data.

4.3 Method

In this section we present our Transductive Parameter Transfer (TPT) method. To point out the generality of the proposed framework, we first introduce the TPT algorithm while the application scenarios chosen for evaluation are described in Sec. 4.4. TPT is a transfer learning technique for parametric classifiers. However, in Sec. 4.3.8 we show that TPT can be easily extended to a semi-parametric framework and in Sec. 4.4 we provide results for both the full-parametric and the semi-parametric versions.

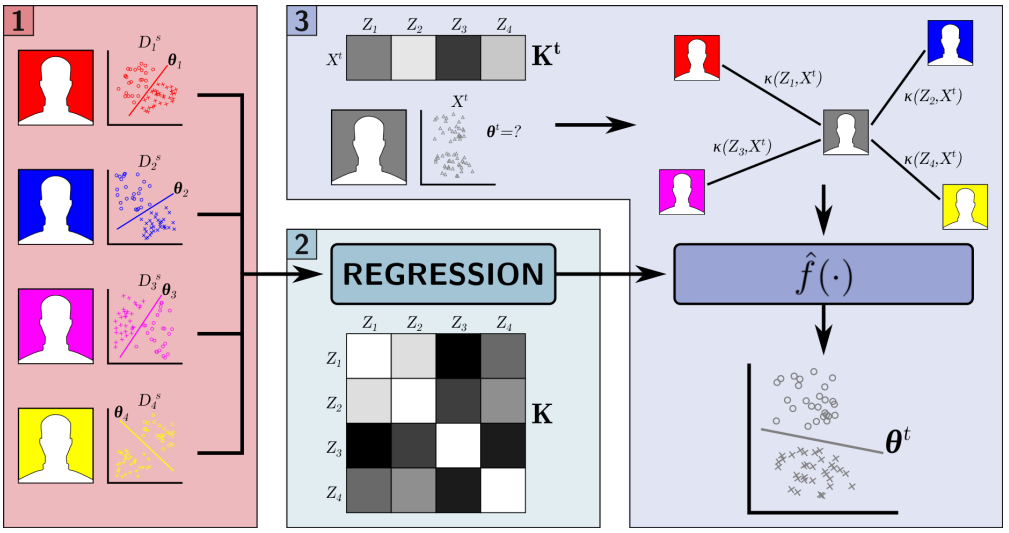


FIGURE 4.2: Overview of the proposed *Transductive Parameter Transfer (TPT)* approach. Box 1: Learning user-specific source classifiers. Box 2: Learning a distribution-to-classifier regression function. Box 3: Computing the target classifier.

4.3.1 Notation and Definitions

Given an *unlabeled target* dataset $X^t = \{\mathbf{x}_j^t\}_{j=1}^{n^t}$ and N *labeled source* datasets $D_1^s, ..., D_N^s$, $D_i^s = \{\mathbf{x}_j^s, y_j^s\}_{j=1}^{n_i^s}$, $\mathbf{x}_j^s, \mathbf{x}_j^t \in \mathcal{X}$, $y_j^s \in \mathcal{Y}$, we want to learn a classifier on the target data X^t without acquiring labeled information. In the context of user personalization D_i^s contains all the training samples corresponding to the i-th source person, while X^t is the set of (unlabeled) data of the target individual for whom we aim to construct the personalized classifier. The feature and the label spaces are denoted as $\mathcal{X}$ and $\mathcal{Y}$, respectively. As we focus on classification $\mathcal{Y} = \{-1, 1\}$ in the binary setting while $\mathcal{Y} = \{1, \ldots, C\}$ in the case of multiple classes. Moreover, $X_i^s = \{\mathbf{x}_j^s\}_{j=1}^{n_i^s}$ denotes the set of points in D_i^s obtained by discarding the label information.

We assume that the vectors in X_i^s are generated by a marginal distribution P_i^s defined on $\mathcal{X}$ and similarly the vectors X^t are generated by P^t. We generally assume that $P^t \neq P_i^s$ and $P_i^s \neq P_j^s$ $(1 \leq i, j \leq N,\ i \neq j)$. Finally, we call $\mathcal{P}$ the space of all possible distributions on $\mathcal{X}$ and we assume that P^t, P_i^s are i.i.d. sampled on $\mathcal{P}$ according to the meta-distribution Π, *i.e.* $P^t, P_i^s \sim \Pi$. In the following $(\cdot)'$ denotes the transpose operator.

4.3.2 Overview

The proposed TPT approach is based on three main phases (Fig. 4.2). In the first phase, N user-specific classifiers are learned, one for each source training set D_i^s. Each classifier is defined by a parameter vector $\boldsymbol{\theta}_i$. Then a regression algorithm is proposed in order to learn the relation between the marginal distributions P_i^s and $\boldsymbol{\theta}_i$. Finally, the desired target classifier $\boldsymbol{\theta}^t$ is obtained by applying the learned distribution-to-classifier mapping and using as input the distribution P^t. In the following, the three phases are further detailed.

4.3.3 Phase 1: Learning User-specific Source Classifiers

In TPT the source datasets D_i^s are used to learn N independent classifiers by solving N different problems:

$$\boldsymbol{\theta}_i = \min_{\boldsymbol{\theta} \in \Theta} \mathcal{R}(\boldsymbol{\theta}) + \lambda_L L(D_i^s; \boldsymbol{\theta}) \tag{4.1}$$

where Θ is the parameter space, $\mathcal{R}(\cdot)$ a regularizer and $L(\cdot)$ is the empirical risk. The parameter λ_L regulates the trade-off between loss and regularization. Each weight vector $\boldsymbol{\theta}_i$ represents a personalized classifier since it is learned using user-specific samples D_i^s. While our framework is general and different choices can be made for the regularization and the loss terms in (4.1), here we consider a set of linear SVMs. Therefore, defining $\mathcal{X} \equiv I\!R^M$, the optimal decision hyperplane $\boldsymbol{\theta}_i = [\mathbf{w}_i',\ b_i]'$, $\mathbf{w}_i \in I\!R^M, b_i \in I\!R$ can be computed by solving:

$$\min_{\mathbf{w},b} \frac{1}{2}||\mathbf{w}||^2 + \lambda_L \sum_{j=1}^{n_i^s} l(\mathbf{w}'\mathbf{x}_j^s + b, y_j^s) \tag{4.2}$$

where $l(y, \hat{y}) = \max(0, 1 - y\hat{y})$ is the hinge loss.

4.3.4 Phase 2: Learning a Distribution-to-Classifier Mapping

In the second phase of TPT, we propose a regression framework in order to learn a mapping $f : \mathcal{P} \rightarrow \Theta$ between a sample distribution and its associated classifier. The intuition is straightforward: if we are able to learn the relationship between the underlying distribution P_i^s and the corresponding hyperplane $\boldsymbol{\theta}_i$, then, when computing the

optimal hyperplane on the target data we do not need labeled samples since we can simply apply the learned mapping $f(\cdot)$, *i.e.* $\boldsymbol{\theta}^t = f(P^t)$. However, since P_i^s $(1 \leq i \leq N)$ and P^t are unknown, we need to approximate these distributions using the empirical data at disposal. In particular in this work we use all the samples in X^t to approximate P^t while for P_i^s we evaluate two possibilities: (i) all the data in X_i^s are considered and (ii) only the Support Vectors obtained by solving (4.2) for each of the i-th source tasks are used as a proxy for P_i^s.

Let $V_i = \{\mathbf{v}_j\}_{j=1}^{m_i}$ be the set of Support Vectors associated with $\boldsymbol{\theta}_i$ $(V_i \subseteq X_i^s)$. The distribution generating V_i is generally different from the distribution generating X_i^s. In fact V_i does not include those points which are far from the decision hyperplane. Thus approximating P_i^s using V_i introduces an error. Anyway, using V_i instead of X_i^s brings two advantages in our regression framework. The first is that there is a well-known relation between the Support Vectors and the decision hyperplane, *i.e.* $\mathbf{w}_i = \sum_{j=1}^{m_i} \alpha_j y_j \mathbf{v}_j$, where y_j is the label associated with $\mathbf{v}_j$ and α_j the corresponding Lagrange multiplier obtained solving (4.2). Note that we do not know the labels neither the support vectors for the target task, thus we cannot directly compute $\mathbf{w}^t$. However there exists a relation between V_i and $\boldsymbol{\theta}_i$, supporting the validity of our regression framework.

The second advantage in using V_i in place of X_i^s is in term of computational cost and storage space, since typically $m_i < n_i^s$. In the following we assume that Z_i is the set of points chosen to approximate P_i^s, either $Z_i = V_i$ or $Z_i = X_i^s$. Given a training set $\mathcal{T} = \{(Z_i, \boldsymbol{\theta}_i)\}_{i=1}^N$ we propose to learn a mapping $\widehat{f} : 2^{\mathcal{X}} \rightarrow \Theta$ which approximates $f(\cdot)$. The function $\widehat{f}(\cdot)$ is a vector-valued set function, *i.e.* a function which takes as input a set X and outputs a vector $\boldsymbol{\theta} \in I\!R^{M+1}$. In this work we investigate two possible approaches to compute $\widehat{f}(\cdot)$, *i.e.* learning $M+1$ independent scalar regressors $\widehat{f}_k(X)$:

$$\widehat{f}(X) = (\widehat{f}_1(X), ..., \widehat{f}_{M+1}(X)) \tag{4.3}$$

and using the multi-output regression framework in [118].

Learning Independent Regression Models. In the case of independent regression models, we compute each $\widehat{f}_k(\cdot)$ using the ϵ-insensitive Support Vector Regression (SVR) framework [119] which, in our setting, can be formulated as follows. Each $\widehat{f}_k(\cdot)$ $(1 \leq$

$k \leq M+1$) is defined by a set of parameters $\boldsymbol{\pi}_k = (\boldsymbol{\beta}_k, c_k)$:

$$\widehat{f}_k(X) = \boldsymbol{\beta}_k' \phi(X) + c_k, \tag{4.4}$$

where $\boldsymbol{\beta}_k$ and c_k is the weight vector and the bias, respectively and $\phi(X)$ is a nonlinear mapping of X to a higher-dimensional space. In turn $\boldsymbol{\pi}_k$ can be found by:

$$\min_{\boldsymbol{\pi}} \frac{1}{2}||\boldsymbol{\beta}_k||^2 + \lambda_E \sum_{i=1}^{N} |\theta_{ik} - \widehat{f}_{\boldsymbol{\pi}}(Z_i)|_\epsilon \tag{4.5}$$

where θ_{ik} is the scalar value corresponding to the k-th dimension of $\boldsymbol{\theta}_i$, $|e|_\epsilon = \max(0, |e| - \epsilon)$ is the ϵ-insensitive loss function, λ_E and ϵ are user-defined parameters.

The problem (4.5) can be solved in its dual form [119]:

$$\begin{aligned} \max_{\delta_i^k} \quad & -\frac{1}{2} \sum_{i,l=1}^{N} \delta_i^k \delta_l^k \kappa(Z_i, Z_l) + \sum_{i=1}^{N} \theta_{ik} \delta_i^k - \epsilon \sum_{i=1}^{N} |\delta_i^k| \\ \text{s.t.} \quad & \sum_{i=1}^{N} \delta_i^k = 0, \;\; |\delta_i^k| \leq \lambda_E \;\; (k = 1, ..., M+1) \end{aligned} \tag{4.6}$$

where δ_i^k is the set of Lagrange multipliers and $\kappa(Z_i, Z_l) = \phi(Z_i)'\phi(Z_l)$ is the kernel function. Note that the same kernel can be used for all $k = 1, ..., M+1$, since $\kappa(Z_i, Z_l)$ estimates the similarity between the sets Z_i and Z_l and is independent from the value of k. In Sec. 4.3.7 we specifically discuss the adopted kernel representations.

In the dual form, (4.4) becomes:

$$\widehat{f}_k(X) = \sum_{i=1}^{N} \delta_i^k \kappa(Z_i, X) + c_k. \tag{4.7}$$

Multi-output Regression. In alternative to learning independent regressors, we also consider the Multioutput Support Vector Regression (M-SVR) framework proposed in [118]. The M-SVR is a generalization of the ϵ-insensitive SVR to a multi-dimensional case. In the M-SVR framework, $\widehat{f}(\cdot)$ can be defined by the parameters $\boldsymbol{\Pi} = (\mathbf{B}, \mathbf{c})$:

$$\widehat{f}(X) = \phi(X)'\mathbf{B} + \mathbf{c}' \tag{4.8}$$

Algorithm 6 The proposed TPT approach.

Input: $D_1^s, ..., D_N^s$, X^t, the parameters λ_L, λ_E, ϵ.

Phase 1. *Learning User-specific Source Classifiers*
Compute $\boldsymbol{\theta}_i$, $\forall i = 1, \ldots, N$ using (4.2).

Phase 2. *Learning a Distribution-to-Classifier Mapping*
Create the training set $\mathcal{T} = \{Z_i, \boldsymbol{\theta}_i\}_{i=1}^N$,
where $Z_i = V_i$ or $Z_i = X_i^s$.
Compute the source kernel matrix $\mathbf{K}$, $\mathbf{K}_{il} = \kappa(Z_i, Z_l)$ using (4.15), (4.16), (4.18) or (4.19).
Given $\mathbf{K}$, $\mathcal{T}$, compute $\widehat{f}(\cdot)$ solving:
$\{M\text{-}SVR\}$ (4.9) or $\{SVR\}$ (4.6) $\forall\ k = 1, ..., M+1$

Phase 3. *Computing the Target Classifier*
Compute the target kernel vector $\mathbf{K}^t$, $\mathbf{K}_i^t = \kappa(Z_i, X^t)$ using (4.15), (4.16), (4.18) or (4.19).
Given $\mathbf{K}^t$, compute $\boldsymbol{\theta}^t = \widehat{f}(X^t)$
using $\{M\text{-}SVR\}$ (4.11) or $\{SVR\}$ (4.7),(4.3).

Output: $\boldsymbol{\theta}^t$

where $\mathbf{B} = [\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_{M+1}]$ and $\mathbf{c} = [c_1, \ldots, c_{M+1}]'$ are the weight matrix and the bias vector, respectively. Similarly to scalar-valued regression, $\boldsymbol{\Pi}$ can be found by:

$$\min_{\boldsymbol{\Pi}} \frac{1}{2} \sum_{i=1}^{M+1} ||\boldsymbol{\beta}_i||^2 + \lambda_E \sum_{i=1}^{N} E(\|\boldsymbol{\theta}_i' - \widehat{f}_{\boldsymbol{\Pi}}(Z_i)\|) \tag{4.9}$$

where $E(\cdot)$ is a loss function which extends to the multi-dimensional case the ϵ-insensitive loss proposed by Vapnik for scalar-valued Support Vector Regression [119], *i.e.*:

$$E(u) = \begin{cases} 0 & u < \epsilon \\ u^2 - 2u\epsilon + \epsilon^2 & u \geq \epsilon \end{cases} \tag{4.10}$$

As for scalar-valued SVR, the problem (4.9) can be solved in its dual form by introducing the kernel matrix $\mathbf{K} \in I\!R^{N \times N}$, $\mathbf{K}_{ij} = \kappa(Z_i, Z_j)$ [118].Hence, the decision function (4.4) can be rewritten as:

$$\widehat{f}(X) = \sum_{i=1}^{N} \boldsymbol{\Delta}_{i\cdot} \kappa(Z_i, X) + \mathbf{c}' \tag{4.11}$$

where $\boldsymbol{\Delta} \in I\!R^{N \times M+1}$ is the matrix of the optimal parameters computed solving the dual optimization problem associated to (4.9) and $\boldsymbol{\Delta}_{i\cdot}$ denotes the i-th row. To compute $\boldsymbol{\Delta}$ and $\mathbf{c}$ in this work we follow [118] and adopt an iterative re-weighted least-squares procedure. This procedure is summarized in Algorithm 7. We define the matrix $\boldsymbol{\Theta} \in I\!R^{N \times M+1}$, $\boldsymbol{\Theta} = [\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N]'$. At each iteration k, the values of $\boldsymbol{\Delta}$ and $\mathbf{c}$ are updated

Algorithm 7 Optimization algorithm to solve (4.9)

Input: The set $\mathcal{T} = \{Z_i, \boldsymbol{\theta}_i\}_{i=1}^N$, the parameters λ_E, ϵ.

Initialize $k = 0$, $\boldsymbol{\Delta}^k = \mathbf{0}$, $\mathbf{c}^k = \mathbf{0}$.
Inner Loop:
Compute a_i using (4.13), $i = 1, \ldots, N$.
Compute $\hat{\boldsymbol{\Delta}}$, $\hat{\mathbf{c}}$ solving (4.12) $\forall\ j = 1, \ldots, M+1$.
Compute η_k using a backtracking algorithm.
Compute $\boldsymbol{\Delta}^{k+1} = \boldsymbol{\Delta}^k + \eta_k(\hat{\boldsymbol{\Delta}} - \boldsymbol{\Delta}^k)$.
Compute $\mathbf{c}^{k+1} = \mathbf{c}^k + \eta_k(\hat{\mathbf{c}} - \mathbf{c}^k)$.
Set $k = k + 1$.
Until Convergence

Output: $\boldsymbol{\Delta}, \mathbf{c}$

solving a series of $M+1$ independent weighted least-squares problems, one for each column of $\boldsymbol{\Delta}$ (here denoted as $\boldsymbol{\Delta}_{\cdot j}$) and for each c_j:

$$\begin{bmatrix} \mathbf{K} + \mathbf{A} & \mathbf{1} \\ \mathbf{a}'\mathbf{K} & \mathbf{1}'\mathbf{a} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Delta}_{\cdot j} \\ c_j \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Theta}_{\cdot j} \\ \mathbf{a}\boldsymbol{\Theta}_{\cdot j} \end{bmatrix} \tag{4.12}$$

where $\boldsymbol{\Theta}_{j\cdot}$ is the j-th row of the matrix $\boldsymbol{\Theta}$ and $\mathbf{1}$ is an all-one column vector. The vector $\mathbf{a} = [a_1, \ldots, a_N]$ and the matrix $\mathbf{A} \in I\!R^{N \times N}$, $\mathbf{A}_{ij} = a_i I(i-j)$, where $I(\cdot)$ is an indicator function, are computed at each step using:

$$a_i = \begin{cases} 0 & u_i^k < \epsilon \\ \frac{2\lambda_E(u_i^k - \epsilon)}{u_i^k} & u_i^k \geq \epsilon \end{cases} \tag{4.13}$$

and $u_i^k = \|\boldsymbol{\theta}_i' - \sum_{i=1}^N \boldsymbol{\Delta}_{i\cdot}^k \kappa(Z_i, X) - (\mathbf{c}^k)'\|$. For more details on the M-SVR framework we refer the reader to the original paper [118].

4.3.5 Phase 3: Computing the Target Classifier

In the last phase of TPT the optimal target classifier $\boldsymbol{\theta}^t$ is computed considering the unlabeled target samples X^t and using $\boldsymbol{\theta}^t = \widehat{f}(X^t)$. For M-SVR we use (4.11), while (4.7) and (4.3) are used in the case of independent regression models. Note that the computations which involve the source data (Phase 1-2) are performed only once. Then, for every new user, only Phase 3 needs to be repeated. This is very advantageous in real world applications, where it is desirable to accomplish personalization in a limited time frame. Algorithm 6 summarizes the main phases of TPT.

4.3.6 Test Phase

Once $\boldsymbol{\theta}^t = [(\mathbf{w}^t)', b^t]'$ has been computed, the test phase is a standard classification with linear SVMs. Given a new target feature vector $\mathbf{x}$, the corresponding label y is predicted as $y = \text{sign}(\mathbf{x}'\mathbf{w}^t + b^t)$. It is worth noting that our TPT framework can be trivially extended to a multi-class setting adopting a one-versus-all scheme.

4.3.7 Kernels for Distributions

From Algorithm 6 it is clear that both in the case of independent regression models and for multi-output regression, the dual optimization problems and the decision functions only depend on the kernel matrix. It is worth noting that $\kappa(\cdot)$ is defined on *sets* of points and not on feature vectors (*i.e.* single data points) as it is more common. The role of the kernel here is to estimate the similarity between distributions, empirically represented by sets of data points. In the following we propose different (non-exhaustive) choices for the kernel function.

4.3.7.1 EMD-based kernel

The Earth Mover's Distance (EMD) [3, 20] represents a simple and practical approach to measure the distance between distributions. To compute the EMD between X_i and X_j, first a clustering algorithm is applied separately to the two datasets (we use a simple k-means algorithm in our experiments). In this way the signatures of each set $\mathcal{I} = \{(\boldsymbol{\nu}_1^i, w_1^i), \dots (\boldsymbol{\nu}_Q^i, w_Q^i)\}$ and $\mathcal{J} = \{(\boldsymbol{\nu}_1^j, w_1^j), \dots (\boldsymbol{\nu}_Q^j, w_Q^j)\}$ are computed, where $\boldsymbol{\nu}_q^i$, $\boldsymbol{\nu}_q^j$ are the cluster centroids respectively obtained on X_i and X_j and w_q^i, w_q^j denote the weights associated to each cluster. In this work we set $Q = 20$ and we use the cardinality of each cluster as the cluster weight.

Given two signatures $\mathcal{I}$ and $\mathcal{J}$, the EMD between the associated datasets X_i and X_j is defined as the solution of the following transportation problem:

$$
\begin{aligned}
D_{EMD}(X_i, X_j) = \quad & \min_{f_{pq} \geq 0} \sum_{p,q=1}^{Q} d_{pq} f_{pq} \\
\text{s.t.} \quad & \sum_{p=1}^{Q} f_{pq} = w_q^i \quad \sum_{q=1}^{Q} f_{pq} = w_p^j
\end{aligned}
\tag{4.14}
$$

where f_{pq} are flow variables and d_{pq} is the ground distance defined as $d_{pq} = \|\boldsymbol{\nu}_p^i - \boldsymbol{\nu}_q^j\|$. In a nutshell, the EMD represents the minimum cost needed to transform one distribution into another. Using EMD we define a kernel:

$$\kappa_{EMD}(X_i, X_j) = e^{-\rho D_{EMD}(X_i, X_j)} \tag{4.15}$$

where ρ is a user defined parameter. Despite this is not a valid kernel as it is not semi-definite positive we observe excellent performance in our experimental evaluation. This is in line with the findings of previous works [120].

4.3.7.2 Fisher Kernel

Fisher kernels [121], originally proposed in statistics and machine learning to measure the similarity between distributions, have recently become common tools in computer vision and multimedia [122].

Suppose that the set of points $X = \{\mathbf{x}_t\}_{t=1}^n$ is generated by the marginal distributions P on $\mathcal{X}$. Let p_γ be a probability density function which models the generative process of the elements in X where γ is the parameter vector governing p_γ. In statistics, the score function is defined as $G_\gamma = \nabla_\gamma \log p_\gamma(X)$, *i.e.* it is the gradient of the log-likelihood of the data with respect to the model parameters and describes how the parameters of the generative model p_γ should be modified to better fit the data [121]. Typically p_γ is chosen as a Gaussian Mixture Model and $\gamma = \{\alpha_h, \mu_h, \Sigma_h\}_{h=1}^H$, being H the number of components and $\alpha_h, \mu_h, \Sigma_h$ the component weight, its mean and its covariance matrix, respectively. In our experiments we set $H = 20$ and we assume that every Σ_h is diagonal, *i.e.* $\Sigma_h = \text{diag}(\sigma_h)$.

Given two sets of points X_i and X_j generated by the two distributions P_i and P_j, their similarity can be measured using the Fisher Kernel [121]:

$$\kappa_{FK}(X_i, X_j) = (G_\gamma^{X_i} E_\gamma)' E_\gamma G_\gamma^{X_j} = (\mathcal{G}_\gamma^{X_i})' \mathcal{G}_\gamma^{X_j} \tag{4.16}$$

where $F_\gamma = E_\gamma' E_\gamma$ is the Cholesky decomposition of the Fisher Information Matrix [121] and $\mathcal{G}_\gamma^X$ is the so called Fisher vector. The Fisher vector [122] is obtained by computing

and concatenating the following terms ($\forall h = 1, \ldots, H$):

$$\begin{aligned}
\mathcal{G}^X_{\alpha_h} &= \frac{1}{\sqrt{\omega_h}} \sum_t (\psi_t(h) - \omega_h) \\
\mathcal{G}^X_{\mu_h} &= \frac{1}{\sqrt{\omega_h}} \sum_t \psi_t(h) \frac{\mathbf{x}_t - \mu_h}{\sigma_h} \\
\mathcal{G}^X_{\sigma_h} &= \frac{1}{\sqrt{2\omega_h}} \sum_t \psi_t(h) \left[\frac{(\mathbf{x}_t - \mu_h)^2}{\sigma_h^2} - 1 \right]
\end{aligned}$$

where $\omega_h = \frac{\exp(\alpha_h)}{\sum_j \exp(\alpha_j)}$ and $\psi_t(h)$ represents the soft assignment of $\mathbf{x}_t$ to the h-th Gaussian.

4.3.7.3 Principal Components Kernel

Assuming that each P_i is a multivariate Gaussian distribution ($P_i = \mathcal{N}(\boldsymbol{\mu}_i, \Sigma_i)$), X_i, drawn from P_i, can be represented by $(\boldsymbol{\mu}_i, V_i)$, where $\boldsymbol{\mu}_i$ is the centroid of X_i and V_i is a column-wise matrix containing the principal eigenvectors extracted from X_i. In the following we indicate with V_{ik} the k-th column of V_i, corresponding to the k-th eigenvector. In constructing V_i we select the eigenvectors corresponding to the 80% of the total energy obtained by summing the associated eigenvalues. Then we define:

$$D_{PCA}(X_i, X_j) = ||\boldsymbol{\mu}_i - \boldsymbol{\mu}_j||_2 + \gamma \sum_k d(V_{ik}, V_{jk}), \tag{4.17}$$

and:

$$\kappa_{PCA}(X_i, X_j) = e^{-\rho D_{PCA}(X_i, X_j)}, \tag{4.18}$$

where $d(V_{ik}, V_{jk})$ is the cosine distance between V_{ik} and V_{jk} and γ is a mixing parameter assessing the relative importance of the two terms, respectively modeling the distance between the centroids of the distributions and the cumulative difference of the eigenvector directions.

Intuitively, (4.17) computes the "alignment" mismatch between X_i and X_j when represented as Gaussian clouds of points. It is worth noting that the assumptions of the PCA-based kernel are much stronger than those of other kernels (*e.g.* the Fisher Kernel, in which each P_i is modeled with a Mixture of Gaussians). Nevertheless, our experimental results in Sec. 4.4 demonstrate that this kernel also provides good recognition accuracy, despite its simplicity.

4.3.7.4 Density Estimate-based Kernel

The last choice for a kernel measuring the similarity of two distributions is taken from [123]. It is based on a Density Estimate (DE) kernel and it is defined as follows:

$$\kappa_{DE}(X_i, X_j) = \frac{1}{nm} \sum_{p=1}^{n} \sum_{q=1}^{m} \kappa_{\mathcal{X}}(\mathbf{x}_p, \mathbf{x}_q), \tag{4.19}$$

where n, m are the cardinality of X_i, X_j, respectively, and $\kappa_{\mathcal{X}}(\cdot)$ is a normalized Gaussian kernel defined on the feature space $\mathcal{X}$.

4.3.8 Extension to Distance Learning

The TPT method so far presented is based on parametric, linear classifiers whose parameter vectors $\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_N$ are used to train a regression function. In this section we show how the proposed approach can be extended to deal with semi-parametric non-linear classifiers. Specifically we consider a Distance Learning (DL) algorithm [22] and we call this extension TPTDL.

In distance learning, k-nearest neighbour is typically used for classification. However, the Euclidean distance between sample points is replaced by a metric, usually parametrized by a matrix $\boldsymbol{A}$ learned with a discriminative criterion [124]. In a multi-class scenario $\boldsymbol{A}$ can be further split in class-specific parameter vectors, each vector defining a class-specific distance function [125].

We show below how TPTDL can be obtained by partially changing the 3 phases of TPT as previously defined. First, in Phase 1 of Algorithm 6 the source-specific classifiers (linear SVMs) are replaced by a set of user-and-class-specific distance functions parametrized by $\boldsymbol{\theta}_i = \boldsymbol{A}_i^s$, where $\boldsymbol{A}_i^s = [\mathbf{a}_{i,1}, \ldots, \mathbf{a}_{i,C}]$ is a column-wise matrix in which every column $\mathbf{a}_{i,c}$ defines the metric associated to *the i-th source user and the c-th class* (recall that we assume C classes in a multi-class scenario). Every $\mathbf{a}_{i,c}$ is learned using a set of triplets $\mathcal{S}_{i,c} = \{(\mathbf{x}, \mathbf{x}^+, \mathbf{x}^-)\}$, extracted from X_i^s (*i.e.* $\mathbf{x}, \mathbf{x}^+, \mathbf{x}^- \in X_i^s$), which satisfy the constraints:

$$d_{i,c}(\mathbf{x}, \mathbf{x}^+) < d_{i,c}(\mathbf{x}, \mathbf{x}^-), \tag{4.20}$$

where $d_{i,c}(\cdot)$ is a distance function which states that $\mathbf{x}$ is closer to $\mathbf{x}^+$ than to $\mathbf{x}^-$ and it is defined as $d_{i,c}(\mathbf{x}_1, \mathbf{x}_2) = |\mathbf{x}_1 - \mathbf{x}_2|'\mathbf{a}_{i,c}$. The triplets in $\mathcal{S}_{i,c}$ are chosen in such a way that $\mathbf{x}$ and $\mathbf{x}^+$ are associated with the same ground truth class label c while $\mathbf{x}^-$ is associated with a class label $y \neq c$. In a multi-class scenario we use class-specific distance functions (*i.e.*, depending on c) because they have been proven to be more effective than a single metric [125]. We refer to [125] for the details concerning how each vector $\mathbf{a}_{i,c}$ can be learned.

Once we obtain a set of user-and-class-specific distance functions, parametrized by $\boldsymbol{\theta}_i, ..., \boldsymbol{\theta}_N$ (Phase 1 of Algorithm 6), we use Phase 2 of Algorithm 6 to compute the source and target kernel matrices ($\mathbf{K}$ and $\mathbf{K}^t$) and the regression function $\widehat{f}(\cdot)$. Then, for every new *target* user, we use her/his unlabeled sample points X^t and Phase 3 of Algorithm 6 to compute the target matrix $\boldsymbol{A}^t = [\mathbf{a}_1^t, \ldots, \mathbf{a}_C^t] = \boldsymbol{\theta}^t = \widehat{f}(X^t)$. We now have a set of class specific metrics $\{d_c^t(\cdot)\}$ for the target user defined by the corresponding column vectors in $\boldsymbol{A}^t$.

At test time we use $\{d_c^t(\cdot)\}$ with a k-nearest neighbour classifier. However, since labels from X^t are not available, we use the labeled samples of the most similar source user, who is selected using the target kernel vector $\mathbf{K}^t$. In other words, we select the source set D_b^s, where $b = \arg\min_{i \in [1,\ldots,N]} \kappa(X_i^s, X^t)$. Hence, in the Test Phase, given a new target feature vector $\mathbf{x}$, we compute its (k-) nearest neighbour(s) solving: $\min_{(\mathbf{x}_j^s, y_j^s) \in D_b^s} d_{y_j^s}^t(\mathbf{x}, \mathbf{x}_j^s)$.

4.4 Experimental Results

We evaluate below the proposed user-personalization approach on three different applications: (i) gesture recognition from smartwatch data, (ii) action unit detection and (iii) pain classification from facial expressions.

As we will see, the first scenario is simpler than the others because of a higher inter-class variability and a lower dimensionality of the adopted feature space. Nevertheless, it is very useful to illustrate the intuitive idea behind the regression-based computation of the personalized classifiers parameters using a two-dimensional projection.

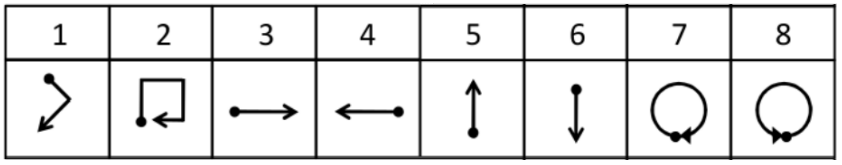


FIGURE 4.3: The set of gestures in the SWGR dataset.

TABLE 4.1: Overall accuracy on the SWGR dataset

Gesture classes	SVM	DL	TPT		TPTDL	
			SVR	M-SVR	SVR	M-SVR
1, 6	88.3	84.3	**90.7**	90.3	86.7	87.0
2, 7	83.0	84.0	88.0	**89.3**	86.9	87.0
3, 4	93.7	96.7	95.2	96.7	96.9	**97.3**
1, 2, 7	88.2	89.6	90.1	**90.2**	90.2	**90.2**
1, 2, 3, 6, 7	86.3	84.9	86.5	86.5	86.4	**86.9**
1-8	87.0	88.5	87.2	87.3	89.9	**90.0**

4.4.1 Smartwatch-based Gesture Recognition

As a first application scenario, we consider the problem of recognizing arm gestures performed by a user wearing an accelerometer-equipped smartwatch. Previous literature [113, 114] has shown that, in general, SVM-based approaches are able to achieve higher accuracy when labeled data from the target user are available as opposite to user-independent solutions.

As collecting labeled gesture samples is an annoying and time consuming process, exploiting TPT in this context can be highly beneficial since it permits to develop a personalized gesture recognition system that completely eschews any configuration phase. In the ideal application scenario this system comes with a pre-loaded generic classifier, trained from a source dataset. This classifier is able to recognize user gestures, although with reduced accuracy. During the initial phase the system does not require any feedback from the user about the correctness of its predictions. Instead, each time the user performs a gesture, the system includes this new data in an unlabeled target dataset. When the target dataset has reached a suitable size, it is used as input to our proposed TPT algorithm, obtaining a personalized classifier which replaces the user-independent one.

We evaluate the performance of TPT on the SmartWatch-based Gesture Recognition (SWGR) dataset presented in [114]. This dataset comprises the 8 gestures shown in Fig. 4.3, each repeated 10 times by 15 different users. For each gesture repetition a manually segmented recording of the smartwatch's 3-axis accelerometer's measurements

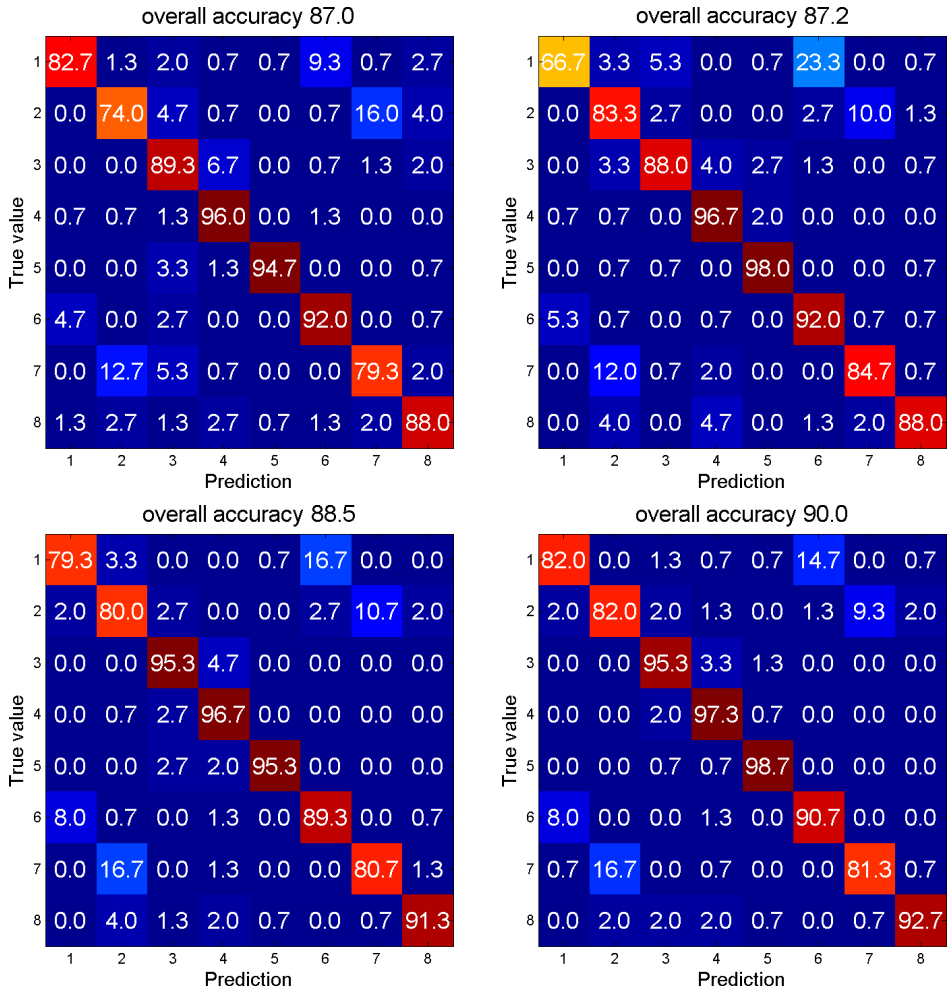
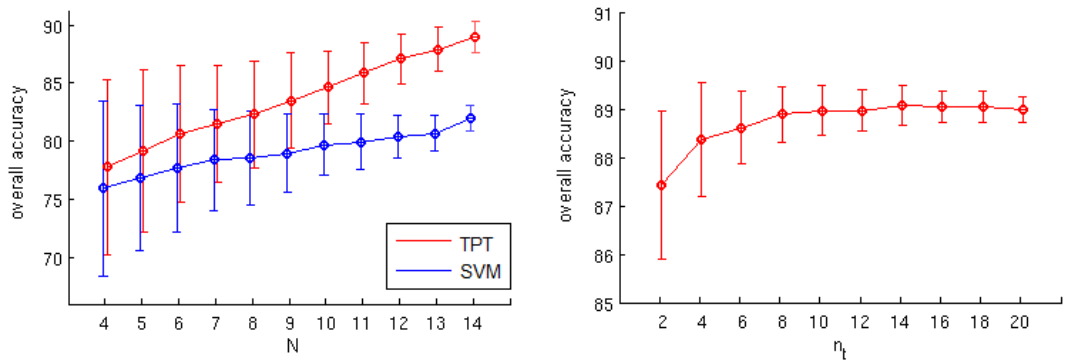


FIGURE 4.4: SWGR dataset. Confusion matrices obtained with (left) a generic classifier and (right) the proposed approach. Top row: SVM and TPT, bottom row: DL and TPTDL.

is given. We consider each of the accelerometer's axes as producing an independent signal, which we process with the Haar Wavelet Transform, as described in [113]. We retain the first 8 coarsest-scale coefficients, and we concatenate them to form a 24-element feature vector which is used to represents data samples.

The results obtained on this dataset are shown in Table 4.1, where the overall accuracy (sum of the diagonal elements of the confusion matrix) achieved at increasing number of classes (*i.e.* $C = 2, 3, 5, 8$) is reported. In the binary classification case ($C = 2$) we chose both pairs of categories which are difficult to discriminate (*e.g.* 2 versus 7 and 1 versus 6) as well as easier classification tasks. In both cases the advantages of our method over user-independent classifiers (either "SVM": linear SVM or "DL": Distance Learning) are evident. In Table 4.1 and in the following we denote with "TPT" the proposed method



(a) (b)

FIGURE 4.5: SWGR dataset (gestures 2,7). Accuracy of TPT (a) at varying number of source users compared with a generic SVM, (b) at varying number of target data points n_t.

when linear SVMs are used as individual classifiers, while "TPTDL" indicates the distance learning-based extension presented in Sec. 4.3.8. Moreover, columns denoted with "SVR" indicate the use of independent regression models while "M-SVR" refers to the multi-output regression framework (see Sec. 4.3.4). Similarly to [114], we note that some classes are more critical to discriminate among each others, such as 2 and 7, or 1 and 6. Note also that, generally speaking, the 2-class task $\{2, 7\}$ is harder than the 3-class task $\{1, 2, 7\}$. This is probably due to the fact that gesture 1 is easier to discriminate with respect to both 2 and 7, which leads to a lower overall error when gesture 1 data points are used for testing together with gestures 2 and 7. A similar phenomenon is observed comparing the all-class task $(1-8)$ with $\{1, 2, 3, 6, 7\}$. With $C > 2$, TPT performs slightly worse than TPTDL, probably because the latter is based on non-linear classifiers which can better model complex data distributions. With $C > 2$, TPTDL also outperforms both SVM and DL. The confusion matrices for the all-class task are shown in Fig. 4.4.

In all the experiments in Table 4.1, TPT is based on the Density Estimate-based Kernel (Sec. 4.3.7) because we observed it guarantees the highest accuracy when the number of data points for each task is limited (see Sec. 4.4.2). In the case of TPTDL we used the Principal Components Kernel. In Table 4.1 we also compare the $M+1$ independent regression models with the M-SVR approach. The advantages of the latter are evident.

To gain a deeper insight of the properties of the proposed personalization method we also analyse in detail one of the binary classification problems: class 2 versus class 7. We apply Principal Component Analysis (PCA) to the features vectors, retaining only the first two principal components (which correspond to about 50% of the total energy).

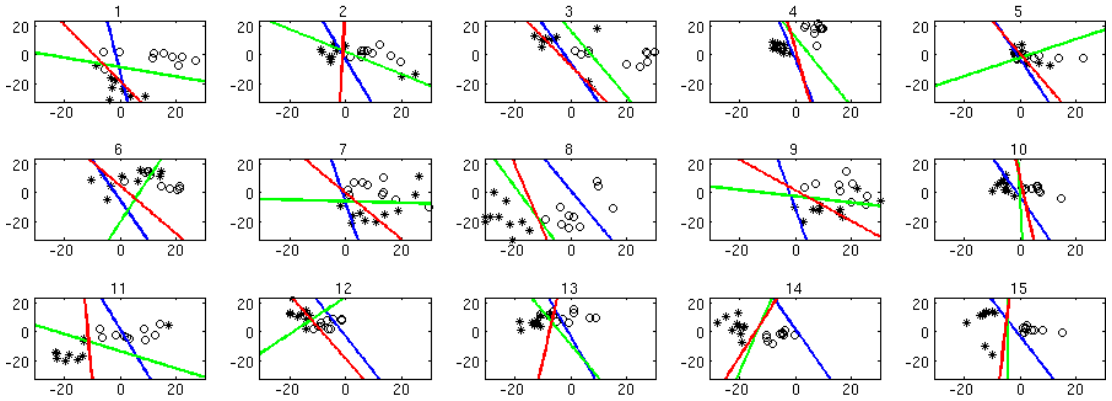


FIGURE 4.6: SWGR dataset. Results for the two-classes case: 2 vs 7. Hyperplanes obtained with (green) ideal, (blue) generic and (red) TPT classifiers.

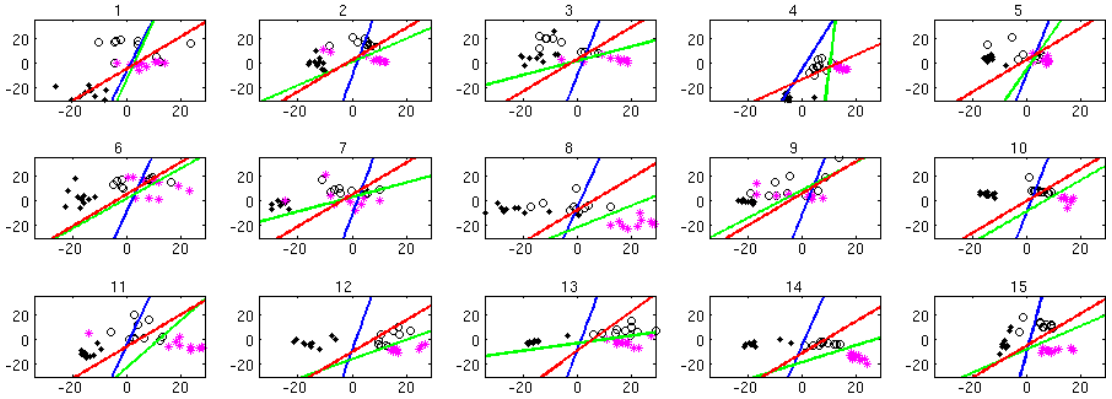


FIGURE 4.7: SWGR dataset. Results for the three-classes case: 1,2 and 7. Hyperplanes obtained with (green) ideal, (blue) generic and (red) TPT classifiers.

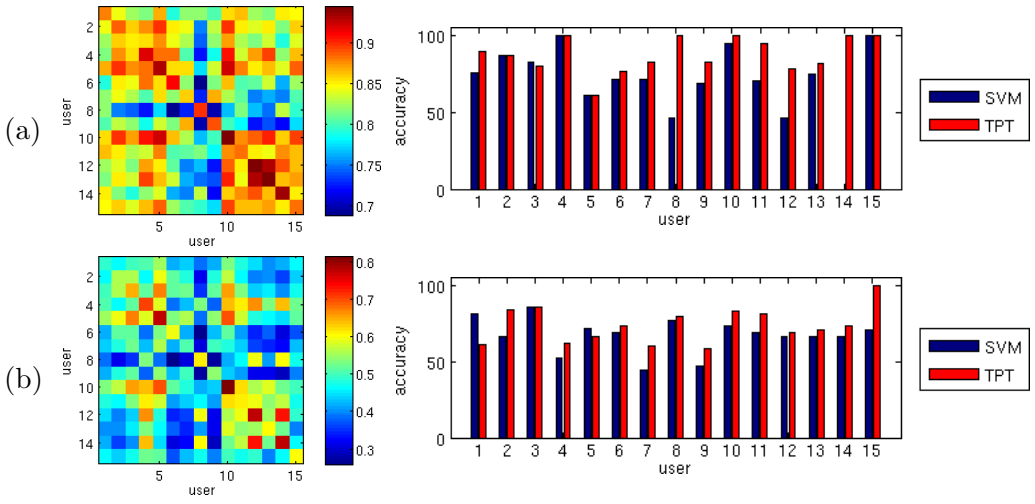


FIGURE 4.8: SWGR dataset. Results for (a) binary (gestures 2 vs 7) and (b) a multiclass classification problems (gestures 1,2,7). (Left) Kernel matrix showing the similarity among the users' data distributions and (right) overall accuracy obtained with a generic SVM classifier (blue) and TPT (red).

The analysis with two dimensional data permits to visually inspect the effects of the distribution mismatch among different individuals. Figure 4.6 shows the projected data points corresponding to gesture type 2 and 7, plotted separately for each of the 15 users. The hyperplanes obtained with a generic classifier (linear SVM trained using the samples of all the source users) and with TPT are plotted respectively in blue and red. The hyperplanes in green are the *ideal* classifiers, *i.e.* those trained using only the target user's *labeled* data (here we adopted the same terminology used in [101]). Note that the hyperplanes of the generic classifiers are slightly different since they are trained with a leave-one-user-out protocol, *i.e.* with slightly different source user sets. Figure 4.6 shows some interesting properties of the classifiers. Firstly, the data points of the two different gesture types performed by the same user are usually quite well separated. However, using a generic classifier for all the users does not seem to be the optimal solution. Indeed, the ideal hyperplanes tend to be very different among them. In most of the cases, the TPT hyperplane separates the two classes better than the generic one, and in same cases (e.g., users 8 and 14) the difference is very pronounced. It is also worth noting that these experiments are based on very few samples per user, which means that the user-specific distributions have been estimated with scarcity of data. In Fig. 4.8(a) the corresponding source kernel matrix and the accuracy obtained with the generic SVM (in blue) and with TPT (in red) are shown for each user. Among all 15 users, the highest improvement is obtained for user 14, followed by user 8, which corresponds to the intuition we get from the visualization of the first 2 components in Fig. 4.6.

Figures 4.7 and 4.8(b) show the results on a three-classes scenario (gestures 1, 2, 7). As the multi-class personalization is performed with a one-vs-all scheme, in Fig. 4.7 we show the classifier learned for one of the three classes, plotted with the symbol "*" and highlighted in pink, while the data points of the other two classes are drawn in black, using two different symbols. The advantage of personalization can still be observed, but compared to the two-classes case it is less evident. In fact in the three-classes case, the data distributions become more complex. As we observed above, in multiclass problems a non-linear classifier such as TPTDL performs better (Table 4.1).

Fig. 4.5(a) shows the performance of our TPT method and a generic SVM classifier (gestures 2-7) at varying number of source users. The x-axis indicates the number N of source users, which have been randomly selected among the 15 users in the SWGR

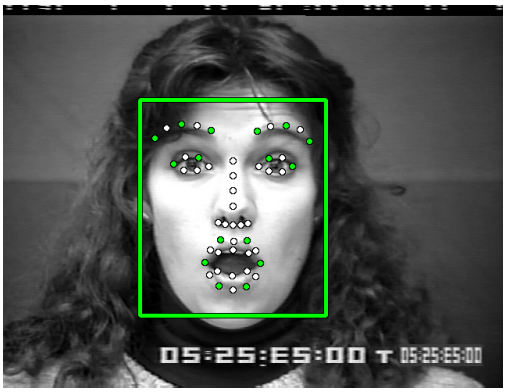


FIGURE 4.9: Sample frame of the CK+ dataset. The green rectangle shows the cropped part of the image and the dots are the detected facial landmarks. The 16 selected landmarks are highlighted in green (better seen at a high magnification).

dataset, repeating the experiment up to 100 runs and finally reporting the average values and the standard deviations (error bars). As expected, the higher N is, the more advantageous TPT is with respect to the generic classifier. In fact, the regression function $\widehat{f}(\cdot)$ (Sec. 4.3.4) is learned using N training samples (*i.e.* data distributions) and, if N is too small, it generalizes poorly.

We also analyse the impact of the number of target data points n_t on the accuracy. For every $n_t \in [2, n]$ (for every user i, $n = |X_i| = 20$ in the 2 classes scenario), we randomly select a target user i and a subset of n_t samples from X_i and we apply TPT to such under-sampled target, obtaining a given accuracy, which is computed *using all the n samples of the target user*. The experiment is repeated up to 100 runs for every value of n_t and the average results are reported in Fig. 4.5(b). Quite surprisingly, only 2 data points are already enough to obtain a high accuracy with the proposed method.

4.4.2 Action Unit Detection

The proposed personalization method has been also applied to the problem of automatic facial Action Unit detection. We use the Extended Cohn Kanade (CK+) dataset [91]. This dataset includes 593 videos from 123 users and contains a set of spontaneous and posed expressions with only frontal faces. The number of videos per user ranges from 1 to 11. The video length varies from 4 to 71 frames. A sample frame from the CK+ dataset is shown in Fig. 4.9.

We follow the pipeline proposed in [101] for feature extraction/representation: first the face and the facial landmarks are detected. The face is then aligned, cropped and resized

TABLE 4.2: Our Method, comparison among different kernels. Performance on CK+ dataset, F_1 Score.

AU	EMD	Fisher	PCA	DE	DE-SVs
1	72.2	74.0	69.5	74.4	**74.9**
2	81.8	75.5	77.3	84.2	**82.4**
4	71.5	71.8	71.3	66.3	**74.2**
6	75.1	74.9	**76.7**	74.8	74.3
12	85.5	83.5	84.4	85.1	**84.6**
17	82.8	83.5	77.7	76.1	**84.3**
Avg	78.2	77.2	76.2	76.8	**79.1**

TABLE 4.3: Our Method, comparison among different kernels. Performance on CK+ dataset, AUC.

AU	EMD	Fisher	PCA	DE	DE-SVs
1	88.0	89.0	86.8	88.2	**89.6**
2	93.5	92.9	92.4	92.6	**93.9**
4	88.1	85.0	85.9	84.3	**88.6**
6	92.2	91.3	**91.5**	91.1	**91.5**
12	**97.5**	97.2	97.4	97.1	**97.5**
17	**95.9**	94.3	92.7	94.3	94.1
Avg	92.5	91.6	91.1	91.3	**92.7**

TABLE 4.4: Comparison among related works. Performance on CK+ dataset. F_1 Score.

AU	SVM	TSVM [100]	KMM [126]	DASVM [127]	STM [101]	TPT DE-SVs
1	61.1	56.8	44.9	57.7	74.0	**74.9**
2	73.5	59.8	50.8	64.3	76.2	**82.4**
4	62.7	51.9	52.3	57.7	69.1	**74.2**
6	75.7	47.8	70.1	68.2	**79.6**	74.3
12	76.7	59.6	74.5	59.0	77.2	**84.6**
17	76.0	61.7	53.2	81.4	**84.3**	**84.3**
Avg	70.9	56.3	57.6	64.7	74.8	**79.1**

TABLE 4.5: Comparison among related works. Performance on CK+ dataset. AUC.

AU	SVM	TSVM [100]	KMM [126]	DASVM [127]	STM [101]	TPT DE-SVs
1	79.8	69.9	68.9	72.6	88.9	**89.6**
2	90.8	69.3	73.5	71.0	87.5	**93.9**
4	74.8	63.4	62.2	79.9	81.1	**88.6**
6	89.7	60.5	87.7	**94.7**	94.0	91.5
12	88.1	76.0	89.5	95.5	92.8	**97.5**
17	90.3	73.1	66.6	94.7	**96.0**	94.1
Avg	85.6	68.7	74.7	83.1	90.1	**92.7**

to 200×200 pixels. Then, 16 landmarks (see Fig. 4.9) are selected and SIFT descriptors are extracted from 36×36 pixels regions around them. Finally, SIFT descriptors are concatenated and dimensionality is reduced using PCA. We retain 90% of the energy, obtaining a final feature vector of size 51. Similarly to [101], we select the most frequent AUs in the dataset and the detection of each AU is considered as an independent binary classification problem. We use the code from [101] available online* for face and facial landmarks detection, and OpenCV for SIFT descriptor extraction. The performance is evaluated in terms of F_1 score, defined as $F_1 = \frac{2 \cdot Recall \cdot Precision}{Recall + Precision}$, and the Area Under ROC (AUC). All the algorithm's parameters (λ_L, λ_E, ϵ) have been set with an inner-loop of cross-validation over the N source users.

The results are shown in Tables 4.2-4.5. We report the performances obtained with our method (TPT) when M-SVR is used and with different kernel choices. We also consider previous methods as baselines: a generic classifier learned on all the source samples (SVM), a semi-supervised Transductive SVM (TSVM) [126] and three transfer learning methods, namely STM [101], Kernel Mean Matching (KMM) [100] and Domain Adaptation SVM (DASVM) [127]. Tables 4.4-4.5 show that those approaches based on personalization achieve higher performance with respect to user-independent ones and

*http://humansensing.cs.cmu.edu/intraface/

that TPT is the most accurate. Finally, comparing different kernels (Tables 4.2-4.3), we observe that the Density Estimate kernel provides the best performance when Support Vectors are used (DE-SVs) to approximate the source data distributions, followed by the EMD kernel computed on all the source samples. We did not report results of experiments using Support Vectors and Fisher and EMD kernels as the performance degrades significantly with respect to DE kernel. We ascribe this behavior to the fact that in the CK+ dataset the number of Support Vectors for each source set is quite limited.

4.4.3 Pain Detection from Facial Expression

As a third application scenario, we tested TPT in the context of pain detection from facial expressions. Automatic pain recognition is of utmost importance for developing HCI solutions for elderly persons. In fact, elderly patients who are cognitively impaired tend to have a decreased ability to communicate and report pain. This often results in the under-detection and under-treatment of pain. We consider the PAINFUL dataset [89], which is composed of 200 video sequences of patients with shoulder injuries. It depicts 25 patients performing a series of active and passive range-of-motion tests with either their affected limb or the unaffected one. The dataset is annotated on a frame basis (48398 frames are labeled by experts using the Prkachin and Solomon Pain Intensity, PSPI, metric system [128]). Example of pain/non pain spontaneous expression, extracted from the dataset, are shown in Fig. 4.1.

We follow the pipeline proposed in [110] for feature extraction/representation. For each frame we use the eye locations provided in the PAINFUL database to crop and warp the face region into a 128×128 pixel image. Then, the resulting face image is divided into 8×8 blocks and Local Binary Pattern Histograms features [90] are extracted on each block. Following the pipeline reported in [110] we adopt $LBP^{u2}_{8,1}$, where *u2* means "uniform" and $(8, 1)$ represents 8 sampling points on a circle of radius 1. The resulting 59-dimensional feature vectors for each block are concatenated resulting into a descriptor of $8 \times 8 \times 59 = 3776$ dimensions. Finally, PCA is applied to reduce feature dimensions retaining 90% of the variance. The dimension of the final feature vectors is 334. Following [110], our experiments are conducted using a leave-one-subject-out evaluation scheme. However, since for one of the subject there are no videos with exhibited pain, we had to exclude

this subject from the training set. Hence, the final number of subjects considered, at training and at testing time, is respectively 24 and 25. The performance is evaluated in terms of AUC. We compare the proposed TPT with M-SVR against a generic classifier (SVM) trained using only the source samples (no domain adaptation), Transductive Transfer Adaboost (TTA) [110], Transductive SVM (TSVM) [126] and Selective Transfer Machine (STM) [101]. For TTA, we report the performance published by Chen *et al.* in [110], while for the last two algorithms, we use the codes publicly available[†,‡]. All the algorithm's parameters (λ_L, λ_E, ϵ) have been set with an inner-loop of cross-validation over the N source users. The results are shown in Table 4.6 [§]. Note that TSVM and STM suffer from the fact that the PAINFUL dataset is strongly unbalanced toward negative samples (no pain frames). For this reason we trained the TSVM and the STM classifiers using different percentages of training data (see Fig. 4.10), obtained equally sampling the data points from the negative and the positive samples sets and we report in Table 4.6 their best results which correspond to 30% of the whole source data points. Note also that, in the case of STM, the training time *for only one target* was over 24 hours (see below), which makes infeasible training with more than 50% of the source data points for a large dataset as PAINFUL.

Similarly to what observed for the CK+ dataset, the best performance is obtained using the DE kernel combined with Support Vectors. However, comparing personalized classifiers with user-independent ones (*i.e.* SVM), we observe that transferring knowledge provides less benefits. We ascribe this fact to the following reasons. First, the PAINFUL dataset is much more difficult than CK+. While in the CK+ all the faces have a frontal pose, in PAINFUL there are large pan and pitch rotations, expressions are spontaneous and inter-individual differences are pronounced. Moreover, in the CK+, only the emotion peaks are annotated (*i.e.* the last frame of each video), while in PAINFUL all the frames are labeled, and the difference between pain and non pain expressions is more subtle. In fact, the pain intensity of positive samples varies from 1 to 16 and these samples are considered all equally positive.

Finally, in the PAINFUL dataset the number of individuals is much lower than in CK+ (24 vs. about 80-90, depending on the specific AU). As shown in Sec. 4.4.1 (see

[†]http://svmlight.joachims.org/

[‡]http://humansensing.cs.cmu.edu/software.html

[§]Note that the results reported here are slightly different from our previous works [7, 8] as we are considering 25 users instead of 24.

TABLE 4.6: Performance on PAINFUL dataset, AUC. (*) Best results obtained using 30% of the data points, see text for details.

SVM	TTA [110]	TSVM* [126]	STM* [101]	TPT			
				EMD	Fisher	DE	DE-SVs
75.6	76.5	73.7	76.7	77.6	77.3	76.6	**78.3**

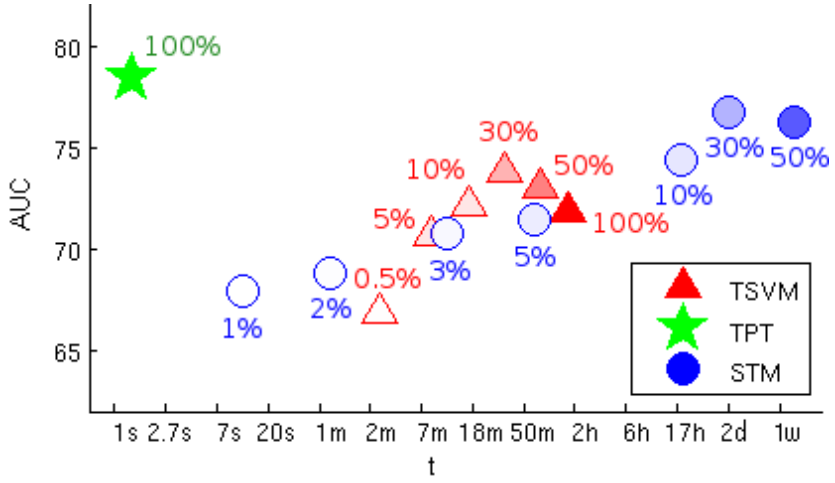


FIGURE 4.10: PAINFUL dataset. AUC vs average time (in logarithmic scale) for training a target classifier for different unsupervised personalization methods. Results for TSVM [126] and STM [101] are reported considering different percentage of source data samples. Our method guarantees the best performance and the shortest training time.

Fig. 4.5(a)), N is a crucial factor for personalization with TPT, being the accuracy of the regression function dependent on the number of source distributions used for training.

The last part of our experimental evaluation is devoted to compare TPT with state-of-the-art approaches in terms of computational times. In Fig. 4.10 we plot the AUC with respect to the training time for TPT and the methods whose code is available on line, *i.e.* TSVM and STM. Note that, for TPT, the only computational time involved in the personalization process with respect to a new target user is Phase 3 of Algorithm 6. Our experiments run on a 4 Cores 2.40GHz CPU machine. In [110] Chen *et al.* reported the training time for TTA on PAINFUL (17.6 minutes) but they did not mention the workstation they used, thus the results are not directly comparable. From Fig. 4.10 it is clear that both TSVM and STM scale poorly as the number of training samples increases. For instance, using only 10% of the source samples, TSVM needs 17 minutes on average for training a personalized classifier, STM 18 hours and TPT only less than a second. Even if the accuracy of all the methods are comparable in this challenging dataset, our approach significantly outperforms all the other algorithms in terms of computational cost.

4.5 Conclusions

We proposed Transductive Parameter Transfer, a framework for building personalized classification models, and we demonstrated its effectiveness on three different application domains: accelerometer-based gesture recognition, pain classification from facial expressions and AU detection. The proposed method is based on a regression framework which is trained to learn the relation between the unlabeled data distribution of a given person and the parameters of her/his personalized classifier. We experimentally showed that our TPT outperforms both user-independent and previous domain adaptation approaches and achieves state of the art performance on public benchmarks. As far as we know, this is the first transductive parameter transfer approach in transfer learning literature. The main advantage of our method is that, using a pre-trained regression function, its computational cost is much lower than other domain adaptation algorithms. This makes TPT an appealing candidate for building personalized systems.

Chapter 5

Discussion and Conclusions

In this thesis, we have addressed the problem of visual data interpretation. The explosive amount of visual data which is continuously being collected entail the accessibility to effective methods for the automatic analysis of this content, in order to allow tasks such as fast data indexing or retrieval. Also, due to the widespread of camera-equipped personal devices and Human Computer Interaction (HCI) applications, the capacity to automatically interpret the users' needs and feedbacks becomes of highest importance, in order to ensure the highest quality of experience to them. This also includes interpreting a user's state based on visual data collected from cameras, for example inferring his/her emotional state (*e.g.* level of happiness, frustration, etc.) from his/her facial expressions.

A significant portion of this overwhelming quantity of visual data originate from CCTV cameras, which daily record continuous streams of images from multiple public locations worldwide. Among these, public scenes on urban scenarios are particularly interesting to analyze, as human activities are depicted in natural settings, as well as challenging due to their typical crowdedness. In this work, we showed that tasks such as automatic high level activities discovery and video summarization can be performed with higher accuracy by considering the correlation among atomic activities in the scene. In particular, we proposed the use of the Earth Mover's Distance to encode this similarity. Still, how to efficiently employ the EMD in our learning framework is not trivial, due to the well known scalability issues of the EMD and the criticalness of correctly defining the distances among visual words, *i.e.* the ground distances of the EMD. In this work, we showed that a higher scalability can be achieved by using an efficient version of the

EMD (EMD-L_1), which reduces the complexity from exponential to linear. In the cases where the histogram bins are naturally sorted, *e.g.* where histogram bins represents contiguous positions in space or time, then EMD-L_1 effectively represents the similarity among the activities associated to these bins. Conversely, for bag-of-words based frameworks, in which each histogram bin corresponds to an atomic activity, we proposed an automatic approach for sorting atomic activities which minimizes the distortion of their original ground distances. We showed that using EMD-L_1 with approximated ground distances still allows to achieve higher performances with respect to using a bin-to-bin distance. Furthermore, we considered the case in which the distances among atomic activities are not defined a priori and we proposed a novel semi-supervised framework which effectively allows to jointly learn these distances and the high level events in the scene. Regardless the discovery of "typical" and "anomalous" behaviors, we showed that different cyclicities of urban life patterns can be highlighted by observing a urban scene at different time granularities: while most of the public datasets depict urban activities in a time range of few hours, thus typically allowing to discover different traffic flows regulated by traffic lights, we showed that typical patterns of urban life shaped by human activities can also be discovered, including daily and weekly patterns and holidays. For this purpose, we released a new dataset recording activities in NYC for over nearly four weeks.

In the work presented in Chapters 2 and 3, we leveraged on low level cues for the discovery of high level behaviors. Still, our framework is very general and it can be possibly applied also in the cases where mid level features or information coming from detectors are considered. The recent advances shown in the context of object detection indeed are heading towards the direction where generic detectors, *i.e.* pre-trained on very large generic datasets, can be used as *off-the-shelf* tools to be applied to a new unseen scene of interest, in order to get the information about the objects in the scene at almost zero cost. Still, due to the known dataset bias problem, the performances of detectors on a new unseen domain tend to be unsatisfactory, thus a domain adaptation step is usually required. Indeed, as a further step to reduce the semantic gap between low level cues and high level behaviors in the analysis of video scenes, the following research directions have also been investigated by the author during her PhD studies: (i) how to self adapt detectors to new unseen domain, for example by leveraging on temporal consistency [11] and (ii) how to fuse low level features with the output of detectors for

video scene interpretation [9, 10].

In the context of model adaptation to new unseen domains, in Chapter 4 we proposed a novel approach (TPT) for obtaining customized models for new unseen target users by leveraging on the similarity shared with other known source users. Besides the advantage of not requiring any additional labeled training samples, this method is especially fast at testing time. This last property is of uttermost importance in contexts like HCI, where the user's quality of experience also depends on the system's response time. The performances of our approach are dependent both on the number of labeled training samples per user and on the number of source users: the first allows enough training data to learn more accurate individual personalized classifiers, the second ensures enough diversity among source domains as well as a more accurate estimation of the distribution-to-parameters mapping function. Most of the domain adaptation problems aim to adapt to a new unseen domain models which are trained on one large size generic dataset, containing an as much varied as possible representation of the considered class. Conversely, in our case we aim to adapt models starting from a large number of individual domains, and build new customized models based on the similarity among the distribution shapes of these domains. Still, while in the case of users expression analysis the identification of the individual domains is straightforward, as each domain correspond to a single user, in other case scenarios such as scene or event recognition, the identification of individual natural domains is not trivial. How to extend our TPT approach to a general domains case scenario is indeed an interesting research direction.

In general, the advances achieved in the context of visual data interpretation in the last years have been remarkable. This was possible thanks to the joint effort from multiple researches within a community aiming to similar goals, sharing ideas, codes and data. Thus, when building on top of previous work (*i.e.* when looking while standing on the giants shoulders) it is important to consider not only the achievements accomplished in terms of methodologies but also in terms of resources such as labeled datasets or pre-learned models. In this sense, domain adaptation and transfer learning approaches play a fundamental role. Still, while in the last decades the task of visual data interpretation has been tackled by focusing mostly on the data itself, the explosive amount of visual data shared and consumed on the web has paved the way to a new learning paradigm. Indeed, everyday millions of users consume multimedia data available on the web and provide feedbacks regarding the content of this data, such as tags, likes, shares, comments, etc.

Differently from dataset labeled by one single expert, this data usually is characterized by incomplete and noisy annotations from multiple users. Still, useful information can be extracted by leveraging on the *wisdom of the crowd*. As future work, an interesting research direction for the author also consists in how to effectively exploit the information gathered from visual data available on the web, in particular in the context of video analysis.

Bibliography

[1] G. Zen and E. Ricci. Earth mover's prototypes: a convex learning approach for discovering activity patterns in dynamic scenes. *CVPR*, 2011.

[2] G. Zen, E. Ricci, and N. Messelodi, S.and Sebe. Sorting atomic activities for discovering spatio-temporal patterns in dynamic scenes. *International Conference on Image Analysis and Processing (ICIAP)*, 2011.

[3] E. Ricci, G. Zen, N. Sebe, and S. Messelodi. A prototype learning framework using EMD: Application to complex scenes analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(3):513–526, 2013.

[4] G. Zen, E. Ricci, and N. Sebe. Exploiting sparse representations for robust analysis of noisy complex video scenes. *ECCV*, 2012.

[5] G. Zen, J. Krumm, N. Sebe, E. Horvitz, and A. Kapoor. Nobody likes mondays: foreground detection and behavioral patterns analysis in complex urban scenes. *ACM/IEEE international workshop on Analysis and Retrieval of Tracked Events and Motion in Imagery Stream (ARTEMIS)*, 2013.

[6] G. Zen, E. Ricci, and N. Sebe. Simultaneous ground metric learning and matrix factorization with earth mover's distance. *IEEE International Conference on Pattern Recognition (ICPR)*, 2014.

[7] E. Sangineto, G. Zen, E. Ricci, and N. Sebe. We are not all equal: Personalizing models for facial expression analysis with transductive parameter transfer. *ACM Multimedia*, 2014.

[8] G. Zen, E. Sangineto, E. Ricci, and N. Sebe. Unsupervised domain adaptation for personalized facial emotion recognition. *ACM International Conference on Multimodal Interaction (ICMI)*, 2014.

[9] G. Zen, N. Rostamzadeh, E. Staiano, J.and Ricci, and N. Sebe. Enhanced semantic descriptors for functional scene categorization. *IEEE International Conference on Pattern Recognition (ICPR)*, 2012.

[10] N. Rostamzadeh, G. Zen, I. Mironică, J. Uijlings, and N. Sebe. Daily living activities recognition via efficient high and low level cues combination and fisher kernel representation. *Image Analysis and Processing (ICIAP)*, 2013.

[11] A. Gaidon, G. Zen, and J. A. Rodriguez. Self-learning camera: Autonomous adaption of object detectors to unlabeled video streams. *arXiv preprint arXiv:1406.4296*, 2014.

[12] A. Torralba and A. A. Efros. Unbiased look at dataset bias. *CVPR*, 2011.

[13] H. Ling and K. Okada. An efficient Earth Mover's Distance algorithm for robust histogram comparison. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(5):840–853, 2006.

[14] D. Bertsimas and J. N. Tsitsiklis. Introduction to linear optimization. 1997. Athena Scientific.

[15] K. Murty. Linear programming. 1983. Wiley, NY.

[16] 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*, 31(3):539–555, 2009.

[17] T. Hospedales, S. Gong, and T. Xiang. A markov clustering topic model for mining behaviour in video. *ICCV*, 2009.

[18] D. Kuettel, M. D. Breitenstein, L. Van Gool, and V. Ferrari. What's going on? Discovering spatio-temporal dependencies in dynamic scenes. *CVPR*, 2010.

[19] J. Li, S. Gong, and T. Xiang. Learning behavioural context. *Int. Journal of Computer Vision (IJCV)*, 97(3):276–304, 2012. ISSN 0920-5691.

[20] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *Int. Journal of Computer Vision (IJCV)*, 40(2):99–121, 2000.

[21] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transaction on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[22] A. Bellet, A. Habrard, and M. Sebban. A survey on metric learning for feature vectors and structured data. *arXiv preprint arXiv:1306.6709*, 2013.

[23] T. Xiang and S. Gong. Video behavior profiling for anomaly detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(5):893–908, 2008.

[24] J. Varadarajan, R. Emonet, and J.-M. Odobez. Probabilistic latent sequential motifs: Discovering temporal activity patterns in video scenes. *BMVC*, 2010.

[25] J. Li, S. Gong, and T. Xiang. Global behaviour inference using probabilistic latent semantic analysis. *BMVC*, 2008.

[26] Y. Yang, J. Liu, and M. Shah. Video scene understanding using multi-scale analysis. *ICCV*, 2009.

[27] C. Niebles, C.-W. Chen, and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. *ECCV*, 2010.

[28] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. *CVPR*, 2009.

[29] M. D. Breitenstein, H. Grabner, and L. Van Gool. Hunting nessie – real-time abnormality detection from webcams. *IEEE Int. Workshop on Visual Surveillance*, 2009.

[30] T. M. Hospedales, J. Li, S. Gong, and T. Xiang. Identifying rare and subtle behaviors: A weakly supervised joint topic model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(12):2451–2464, 2011.

[31] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, 41(3):1–58, 2009.

[32] J. Tang, Z. Chen, A.W. Fu, and D.W. Cheung. Enhancing effectiveness of outlier detections for low density patterns. 2002. Advances in Knowledge Discovery and Data Mining.

[33] M. Breunig, H. Kriegel, R. Ng, and J. Sander. LOF: Identifying density-based local outliers. 2000. ACM SIGMOD International Conference on Management of Data.

[34] J. Friedman, T. Hastie, H. Hofling, and R. Tibshirani. Pathwise coordinate optimization. *Annals of Applied Statistics*, 1:302–332, 2007.

[35] Y. Yao and Y. Lee. Another look at linear programming for feature selection via methods of regularization. 2007. Techn. Report 800, Dept. of Statistics, Ohio State University.

[36] R. Sandler and M. Lindenbaum. Nonnegative matrix factorization with earth mover's distance metric. *CVPR*, 2009.

[37] J Wagner and B. Ommer. Efficiently clustering earth mover's distance. *Asian Conference on Computer Vision (ACCV)*, 2010.

[38] D. Applegate, T. Dasu, S. Krishnan, and S. Urbanek. Unsupervised clustering of multidimensional distributions using earth mover distance. *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2011.

[39] S. Nowozin and S. Jegelka. Solution stability in linear programming relaxations: Graph partitioning and unsupervised learning. *ICML*, 2009.

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

[41] Y. Takano, Y.and Yamamoto. Metric-preserving reduction of earth mover's distance. *Asia-Pacific journal of operational research*, 27(01):39–54, 2010.

[42] O. Pele and M. Werman. A linear time histogram metric for improved sift matching. *ECCV*, 2008.

[43] O. Pele and M. Werman. Fast and robust Earth Mover's Distances. *ICCV*, 2009.

[44] L.I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Phys.*, 60:259–268, 1992.

[45] J. Li, S. Gong, and T. Xiang. Scene segmentation for behaviour correlation. *ECCV*, 2008.

[46] L. Van Gool F. Nater, H. Grabner. Temporal relations in videos for unsupervised activity analysis. *BMVC*, 2011.

[47] S. Shirdhonkar and D. W. Jacobs. Approximate earth mover's distance in linear time. *CVPR*, 2008.

[48] D.D. Lee and H.S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(1):788–791, 1999.

[49] T. Mauthner, P. M. Roth, and H. Bischof. Instant action recognition. *Image Analysis*, pages 1–10, 2009.

[50] K. W. Wilson, B. Raj, P. Smaragdis, and A. Divakaran. Speech denoising using nonnegative matrix factorization with priors. *ICASSP*, pages 4029–4032, 2008.

[51] N. Jiang, K. B. Englehart, and P. A. Parker. Extracting simultaneous and proportional neural control information for multiple-dof prostheses from the surface electromyographic signal. *IEEE Transactions on Biomedical Engineering*, 56(4): 1070–1080, 2009.

[52] A. Cichocki, R. Zdunek, Anh H. Phan, and S. Amari. *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation.* John Wiley & Sons, 2009.

[53] Y. Zhang and B. Li. Particle sorting using a subwavelength optical fiber. *Laser & Photonics Reviews*, 7(2):289–296, 2013.

[54] M. Bilenko and R..J. Basu, S.and Mooney. Integrating constraints and metric learning in semi-supervised clustering. *International conference on Machine Learning*, page 11, 2004.

[55] M. Cuturi and D. Avis. Ground metric learning. *The Journal of Machine Learning Research*, 15(1):533–564, 2014.

[56] N. Jacobs, N. Roman, and R. Pless. Consistent temporal variations in many outdoor scenes. *CVPR*, 2007.

[57] A. Abrams, J. Tucek, J. Little, N. Jacobs, and R. Pless. Lost: Longterm Observation of Scenes (with Tracks). *IEEE Workshop on Applications of Computer Vision (WACV)*, 2012.

[58] T. Bouwmans. Recent advanced statistical background modeling for foreground detection: A systematic survey. *Recent Patents on Computer Science*, 4(3):147–176, 2011.

[59] S. Brutzer, B. Hoferlin, and G. Heidemann. Evaluation of background subtraction techniques for video surveillance. *CVPR*, 2011.

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

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

[62] E. J. Candes, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of ACM*, 58(1):1–37, 2009.

[63] V. Cevher, C. Hegde, M. F. Duarte, and R. G. Baraniuk. Sparse signal recovery using markov random fields. *NIPS*, 2007.

[64] V. Cevher, A. Sankaranarayanan, M. Duarte, D. Reddy, R. Baraniuk, and R. Chellappa. Compressive sensing for background subtraction. *ECCV*, 2008.

[65] X. Cui, J. Huang, S. Zhang, and D. Metaxas. Background subtraction using group sparsity and low rank constraint. *ECCV*, 2012.

[66] M. Dikmen and T. S. Huang. Robust estimation of foreground in surveillance videos by sparse error estimation. *ICPR*, 2008.

[67] C. Zhao, X. Wang, and W. Kuen Cham. Background subtraction via robust dictionary learning. *EURASIP J. Image and Video Processing*, 2011.

[68] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *CVPR*, 2005.

[69] B. Han and L. S. Davis. Density-based multifeature background subtraction with support vector machine. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34:1017–1023, 2012.

[70] V. Reddy, C. Sanderson, and B. C. Lovell. Improved foreground detection via block-based classifier cascade with probabilistic decision integration. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(1):83–93, 2013.

[71] J. Yao and J.-M. Odobez. Multi-layer background subtraction based on color and texture. *CVPR Visual Surveillance workshop (CVPR-VS)*, 2007.

[72] J. Ngiam, C. Y. Foo, Y. Mai, C. Suen, and A. Ng. Unsupervised feature learning and deep learning tutorial. *http://ufldl.stanford.edu/wiki/index.php/UFLDL_Tutorial*.

[73] L. Bo, X. Ren, and D. Fox. Unsupervised feature learning for rgb-d based object recognition. *International Symposium on Experimental Robotics, (ISER)*, 2012.

[74] R. Socher, B. Huval, B. Bhat, C. D. Manning, and A. Y. Ng. Convolutional-recursive deep learning for 3d object classification. *NIPS*, 2012.

[75] M.D. Zeiler, G.W. Taylor, and R. Fergus. Adaptive deconvolutional networks for mid and high level feature learning. *ICCV*, 2011.

[76] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng. Self-taught learning: Transfer learning from unlabeled data. *ICML*, 2007.

[77] S. Rifai, G. Mesnil, P. Vincent, X. Muller, Y. Bengio, Y. Dauphin, and X. Glorot. Higher order contractive auto-encoder. *ECML*, 2011.

[78] S. Rifai, Y. Bengio, A. Courville, P. Vincent, and M. Mirza. Disentangling factors of variation for facial expression recognition. *ECCV*, 2012.

[79] P. Matikainen, M. Hebert, and R. Sukthankar. Action recognition through the motion analysis of tracked features. *ICCV workshop on Video-oriented Object and Event Classification*, 2009.

[80] S. Birchfield. KLT: An implementation of the Kanade-Lucas-Tomasi feature tracker. 2007.

[81] M. Heiler and C. Schnörr. Learning sparse representations by non-negative matrix factorization and sequential cone programming. *The Journal of Machine Learning Research*, 7:1385–1407, 2006.

[82] P. O. Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5:1457–1469, 2004.

[83] H. Tuy. Convex programs with an additional reverse convex constraint. *Journal of Optimization Theory and Applications*, 52(1):463–486, 1987.

[84] J. Yang, K. Yu, and T. Huang. Efficient highly over-complete sparse coding using a mixture model. *ECCV*, 2010.

[85] G. Yu, G. Sapiro, and S. Mallat. Solving inverse problems with piecewise linear estimators: from gaussian mixture models to structured sparsity. *IEEE Transactions on Image Processing*, 21(5):2481–2499, 2012.

[86] I. J. Goodfellow, Q. V. Le, A. M. Sav.e, H. L, and A. Y. Ng. Measuring invariance in deep networks. *NIPS*, 2009.

[87] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflowers: Principles and practise of background maintainance. *ICCV*, 1999.

[88] Z. Zeng, M. Pantic, G. I. Roisman, and T. S. Huang. A survey of affect recognition methods: Audio, visual, and spontaneous expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(1):39–58, 2009.

[89] P. Lucey, J. F. Cohn, K. M. Prkachin, P. E Solomon, S. Chew, and I. Matthews. Painful monitoring: Automatic pain monitoring using the unbc-mcmaster shoulder pain expression archive database. *Image and Vision Computing*, 30(3), 2012.

[90] T. Ahonen, A. Hadid, and M. Pietikainen. Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):2037–2041, 2006.

[91] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews. The extended Cohn-Kanade dataset (CK+): A complete dataset for action unit and emotion-specified expression. *CVPRW*, 2010.

[92] G. Costante, L. Porzi, O. Lanz, P. Valigi, and E. Ricci. Personalizing a smartwatch-based gesture interface with transfer learning. *EUSIPCO*, 2014.

[93] J. Liu, L. Zhong, J. Wickramasuriya, and V. Vasudevan. uwave: Accelerometer-based personalized gesture recognition and its applications. *Pervasive and Mobile Computing*, 5(6):657–675, 2009.

[94] S. D. Roy, T. Mei, W. Zeng, and S. Li. Towards cross-domain learning for social video popularity prediction. *IEEE Transactions on Multimedia*, 15(6):1255–1267, 2013.

[95] S. Xia, M. Shao, J. Luo, and Y. Fu. Understanding kin relationships in a photo. *IEEE Transactions on Multimedia*, 14(4):1046–1056, 2012.

[96] Zhenyu Guo and Z Jane Wang. An unsupervised hierarchical feature learning framework for one-shot image recognition. *IEEE Transactions on Multimedia*, 15 (3):621–632, 2013.

[97] J. Yang, R. Yan, and A. G. Hauptmann. Adapting svm classifiers to data with shifted distributions. *Int. Conf. on Data Mining Workshops*, 2007.

[98] Hal Daumé III. Frustratingly easy domain adaptation. *ACL*, 2007.

[99] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. *CVPR*, 2009.

[100] A. Gretton, A.and Smola, J. Huang, M. Schmittfull, K. Borgwardt, and B. Schölkopf. Covariate shift by kernel mean matching. *Dataset shift in machine learning*, 3(4):5, 2009.

[101] W. S. Chu, F. De La Torre, and J. F. Cohn. Selective transfer machine for personalized facial action unit detection. *CVPR*, 2013.

[102] A. Martinez and S. Du. A model of the perception of facial expressions of emotion by humans: Research overview and perspectives. *J. of Machine Learning Research*, 13(1):1589–1608, 2012.

[103] E. Sangineto. Pose and expression independent facial landmark localization using dense-surf and the hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(3):624–638, 2013.

[104] M. Yeasin, B. Bullot, and R. Sharma. Recognition of facial expressions and measurement of levels of interest from video. *IEEE Transanctions on Multimedia*, 8 (3):500–508, 2006.

[105] M. F. Valstar, B. Jiang, M. Mehu, M. Pantic, and K. Scherer. The first facial expression recognition and analysis challenge. *IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, 2011.

[106] H. Dibeklioğlu, T. Gevers, A. A. Salah, and R. Valenti. A smile can reveal your age: Enabling facial dynamics in age estimation. *ACM Multimedia*, 2012.

[107] G. C. Littlewort, M. S. Bartlett, and K. Lee. Faces of pain: automated measurement of spontaneousallfacial expressions of genuine and posed pain. *ACM International Conference on Multimodal Interfaces (ICMI)*, 2007.

[108] M. F. Valstar, M. Pantic, Z. Ambadar, and J. F. Cohn. Spontaneous vs. posed facial behavior: automatic analysis of brow actions. *ACM International Conference on Multimodal Interfaces (ICMI)*, 2006.

[109] K. Sikka, A. Dhall, and M. Bartlett. Weakly supervised pain localization using multiple instance learning. *IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, 2013.

[110] X. Chen, J.and Liu, P. Tu, and A. Aragones. Learning person-specific models for facial expression and action unit recognition. *Pattern Recognition Letters*, 34(15): 1964–1970, 2013.

[111] S. Mitra and T. Acharya. Gesture recognition: A survey. *IEEE Transactions on Systems, Man, Cybernetics, Part C: Applications and Reviews*, 37(3):311–324, 2007.

[112] G. Bieber, T. Kirste, and B. Urban. Ambient interaction by smart watches. *Int. Conf. on PErvasive Technologies Related to Assistive Environments*, 2012.

[113] M. Khan, S. I. Ahamed, M. Rahman, and J. J. Yang. Gesthaar: An accelerometer-based gesture recognition method and its application in nui driven pervasive healthcare. *Int. Conf. on Emerging Signal Processing Applications (ESPA)*, 2012.

[114] L. Porzi, S. Messelodi, C. M. Modena, and E. Ricci. A smart watch-based gesture recognition system for assisting people with visual impairments. *Int. Workshop on Interactive Multimedia on Mobile & Portable Devices*, 2013.

[115] H. Junker, O. Amft, P. Lukowicz, and G. Tröster. Gesture spotting with body-worn inertial sensors to detect user activities. *Pattern Recognition*, 41(6):2010–2024, 2008.

[116] J. Mäntyjärvi, J. Kela, P. Korpipää, and S. Kallio. Enabling fast and effortless customisation in accelerometer based gesture interaction. *Int. Conf. on Mobile and Ubiquitous Multimedia*, 2004.

[117] S.-J. Cho, E. Choi, W.-C. Bang, J. Yang, J. Sohn, D. Y. Kim, Y.-B. Lee, S. Kim, et al. Two-stage recognition of raw acceleration signals for 3-d gesture-understanding cell phones. *Int. Workshop on Frontiers in Handwriting Recognition*, 2006.

[118] D. Tuia, J. Verrelst, L. Alonso, F. Pérez-Cruz, and G. Camps-Valls. Multioutput support vector regression for remote sensing biophysical parameter estimation. *IEEE Geoscience and Remote Sensing Letters*, 8(4):804–808, 2011.

[119] H. Drucker, C.J.C. Burges, L. Kaufman, A. Smola, and V. Vapnik. Support vector regression machines. *NIPS*, 1997.

[120] M. R. Daliri. Kernel earth mover's distance for eeg classification. *Clinical EEG and neuroscience*, 44(3):182–187, 2013.

[121] T. Jaakkola, D. Haussler, et al. Exploiting generative models in discriminative classifiers. *NIPS*, 1999.

[122] F. Perronnin and C. Dance. Fisher kernels on visual vocabularies for image categorization. *CVPR*, 2007.

[123] G. Blanchard, G. Lee, and C. Scott. Generalizing from several related classification tasks to a new unlabeled sample. *NIPS*, 2011.

[124] E. P. Xing, M. I. Jordan, S. Russell, and A. Y. Ng. Distance metric learning with application to clustering with side-information. *NIPS*, 2002.

[125] K. Q. Weinberger and L. K. Saul. Fast solvers and efficient implementations for distance metric learning. *ICML*, 2008.

[126] T. Joachims. Transductive inference for text classification using support vector machines. *ICML*, 1999.

[127] L. Bruzzone and M. Marconcini. Domain adaptation problems: A dasvm classification technique and a circular validation strategy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(5):770–787, 2010.

[128] K. M. Prkachin and P. E. Solomon. The structure, reliability and validity of pain expression: Evidence from patients with shoulder pain. *Pain*, 139(2):267–274, 2008.