Probabilistic Topic Models for Human Emotion Analysis

by

Prasanth Lade

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved February 2015 by the
Graduate Supervisory Committee:

Sethuraman Panchanathan, Chair
Hasan Davulcu
Baoxin Li
Vineeth N Balasubramanian

ARIZONA STATE UNIVERSITY

May 2015

ABSTRACT

While discrete emotions like joy, anger, disgust etc. are quite popular, continuous emotion dimensions like arousal and valence are gaining popularity within the research community due to an increase in the availability of datasets annotated with these emotions. Unlike the discrete emotions, continuous emotions allow modeling of subtle and complex affect dimensions but are difficult to predict.

Dimension reduction techniques form the core of emotion recognition systems and help create a new feature space that is more helpful in predicting emotions. But these techniques do not necessarily guarantee a better predictive capability as most of them are unsupervised, especially in regression learning. In emotion recognition literature, supervised dimension reduction techniques have not been explored much and in this work a solution is provided through probabilistic topic models. Topic models provide a strong probabilistic framework to embed new learning paradigms and modalities. In this thesis, the graphical structure of Latent Dirichlet Allocation has been explored and new models tuned to emotion recognition and change detection have been built.

In this work, it has been shown that the double mixture structure of topic models helps 1) to visualize feature patterns, and 2) to project features onto a topic simplex that is more predictive of human emotions, when compared to popular techniques like PCA and KernelPCA. Traditionally, topic models have been used on quantized features but in this work, a continuous topic model called the Dirichlet Gaussian Mixture model has been proposed. Evaluation of DGMM has shown that while modeling videos, performance of LDA models can be replicated even without quantizing the features. Until now, topic models have not been explored in a supervised context of video analysis and thus a Regularized supervised topic model (RSLDA) that models video and audio features is introduced. RSLDA learning algorithm performs both dimension reduction and regularized linear regression simultaneously, and has outperformed supervised dimension reduction techniques like SPCA and Correlation based feature selection algorithms. In a first of its kind, two new topic models, Adaptive temporal topic model (ATTM) and SLDA for change detection (SLDACD) have been developed for predicting concept drift in time series data. These models do not assume independence of consecutive frames and outperform traditional topic models in detecting local and global changes respectively.

I dedicate this thesis to my parents for their love, support and guidance.

ACKNOWLEDGEMENTS

This PhD research and the thesis would not have been possible without the support and guidance of my advisor, Professor Sethuraman Panchanathan. It was his trust and encouragement that has given me inspiration to take each step in my research. I thank him for all the knowledge he provided through his interaction and I learnt a lot from his expertise in handling multiple portfolios. His never ending enthusiasm and thirst for knowledge has inspired me to push myself each day and I have no words to express my gratitude to Dr Panch for believing in me and ensuring a successful completion of my thesis.

Dr Vineeth N Balasubramanian played a great role in inspiring me to do research and pursue PhD and I will ever be grateful to him for guiding me when I was transitioning from industry to academia. He has been my PhD mentor and gave me very critical and realistic assessment of my results and ideas. His knowledge in Machine Learning has shown me a path that I treaded, I should say successfully.

I would like to thank Dr Troy McDaniel, who as the Associate Director of the Center for Cognitive Computing (CUbiC) has taught me so much throw his dedication, work ethics and zeal. My committee members Prof Baoxin Li and Prof Hasan Davulcu have helped me learn so much through their inputs and interactions during my research and I am always thankful to them. I would like to thank Dr C K Narayanan who mentored me during the first few years of my PhD and his approach to research and critical thinking helped me throughout my research.

I thank Dr Subhajit Datta who was my mentor at IBM Research who introduced me to this area of research and am very thankful to him for his guidance and collaboration.

I owe a lot to all the members and peers at my CUbiC center especially to Dr Shayok Chakroborty and Dr Rita Chattopadhyay whose PhD work has inspired me as

their junior in the lab. I am happy to know and learn from each of my co-researchers Arash Tadayon, Ramin Tadayon, Hiranmayi Ranganathan, Shantanu Bala, Derrick Rahbar, Bijan Fakhri and Dr Sreekar Krishna. My room mates and PhD students Sai Pavan and Hemanth Kumar have been my continuous support for past 4 years and I am so happy to have friends who personally supported me and I owe them both so much for my PhD research and personal well being.

I would also like to specifically thank all the undergraduate student researchers, Derrick Rahbar, Amy Baldwin, Brandan Jeter, Rebecca Napper, Chaley Boreland and Michael Duran for their contribution to the Social Interaction Assistant project.

This research has been supported by the National Science Foundation (NSF) under Grant No. 1116360. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF.

Last but not the least, all of this cannot be possible without the loving support of my family, my parents who knew the value of education and have always put my aspiration before their necessities and wants. My father's values and guidance helped me retain my moral composure and my mother's love provided comfort. I am thankful to my brother Srikanth and his wife Swetha whose warmth and love played a great role in my life and my research.

TABLE OF CONTENTS

CHAPTER Page

APPENDIX

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

Chapter 1

INTRODUCTION

1.1 Motivation

Medical diagnosis of patients, federal investigations, human-computer interactions and many other applications need an understanding of human emotions. This research broadly addresses the application of social interaction assistance for people who are blind. This needs pattern recognition algorithms that can extract discernible features from video data, learn their relationship with emotions, predict the appropriate emotional state and deliver the prediction to the visually impaired. Human facial expression recognition systems have long relied on discovering basic facial movement patterns that can in turn explain emotions. E.g. Facial Action Units [23] are one such set of facial patterns manually coded and have been used to decipher the emotional states. When a blind user interacts with her peers, she may be interested not only in the exact behavioral state but also in the actual changes in facial patterns. The motivation for this dissertation is to aim for two birds at one shot, where we could extract both emotions and facial changes at the same time.

Happy, sad, disgust, surprise, contempt and anger are most widely accepted affective states and will be called discrete emotions throughout this document. In recent years four new dimensions of emotions have gained popularity viz. arousal (energy), valence (positivity), power (dominance) and expectancy (anticipation) (Figure 1.1 gives more details). Unlike discrete emotions, each of these dimensions can be assigned a real number within a given range and thus are called continuous dimensions. Arousal and valence in fact can be used to define most of the discrete emotions e.g. a high

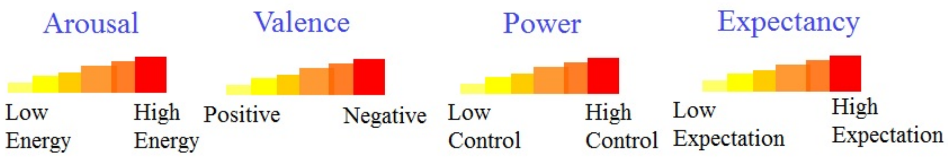


Figure 1.1: Continuous emotion dimensions and what they represent

arousal and valence indicates happiness.

Recognizing both discrete and continuous emotions involves prediction of the state or dimension value at each time step of a video while Emotion change detection (concept drift) is prediction of time steps where prominent changes in emotion occur. These two applications are the focus of discussion and evaluation in this work. To address both these problems we need algorithms that can extract meaningful features and can learn an association between features and emotions. In our search for models that can extract the latent facial patterns, we have come across Probabilistic topic models which are used to extract latent patterns in text documents. At the outset, it looks unreasonable for a novice as to how topic models may be related to problem at hand and the answer lies in the graphical structure of topic models. If we analyze the problem we are trying to solve, it is two fold, 1) to extract facial patterns, and 2) to learn predictive features from video data. This fits well into the framework of topic models because they are double mixture models, where one mixture addresses the first problem and the second mixture solves the second problem.

1.2 Probabilistic Topic Models

Topic models are clustering techniques that have originated to solve the problem of text document retrieval. Text documents contain lots of features, and finding similarity between them is useful to cluster them and also find inherent *topics* that are common to them. In order to cluster documents, they are represented as frequency vectors of terms that occur and the matrix of all document features is called a *document-frequency* matrix. Latent Semantic Indexing models were the first topic models formulated by Deerwester [20] where he proposed an Singular Value Decomposition over the document-term matrix. Thus LSI model can be perceived as a feature projection or dimension reduction technique. Even though LSI models have been very much able to improve document retrieval, they do not work well when new documents containing new topics are to be clustered. They also do not perform well for supervised learning like classifying documents, reviews etc. This led to the formulation of a probabilistic extension to LSI called the probabilistic LSI (pLSI) by Hoffman [27]. Hoffman provided a probabilistic graphical structure to LSI model and showed that the double mixture structure of pLSI is powerful when compared to LSI model. Even though pLSI model provided a probabilistic learning framework, it had some of the disadvantages of LSI with respect to unseen documents. David Blei et al. [13] have proposed Latent Dirichlet Allocation which forms the basis of many of the current topic models. LDA model provided two different insights, 1) how features group to form topics 2) how topics define a document.

Now taking forward our discussion to the context of emotion recognition we illustrate it with diagram. The application of topic models to emotion recognition is an extension to their application to image analysis. In Figure 1.2, we have shown a skying image and let us assume that the image features have been quantized. If a human being were to annotate the image , then they would be *human*, *trees*, *snow* , *sky* etc. Topic models precisely do this where they annotate an image with *topics* that occur and they also define how a *topic* is defined in terms of features. Similarly we hypothesize that topic models can model audio and image frames where they assign topics to images and these topics are defined in terms of the audio and video features. The first insight into topics per image can be used to predict emotions and the second

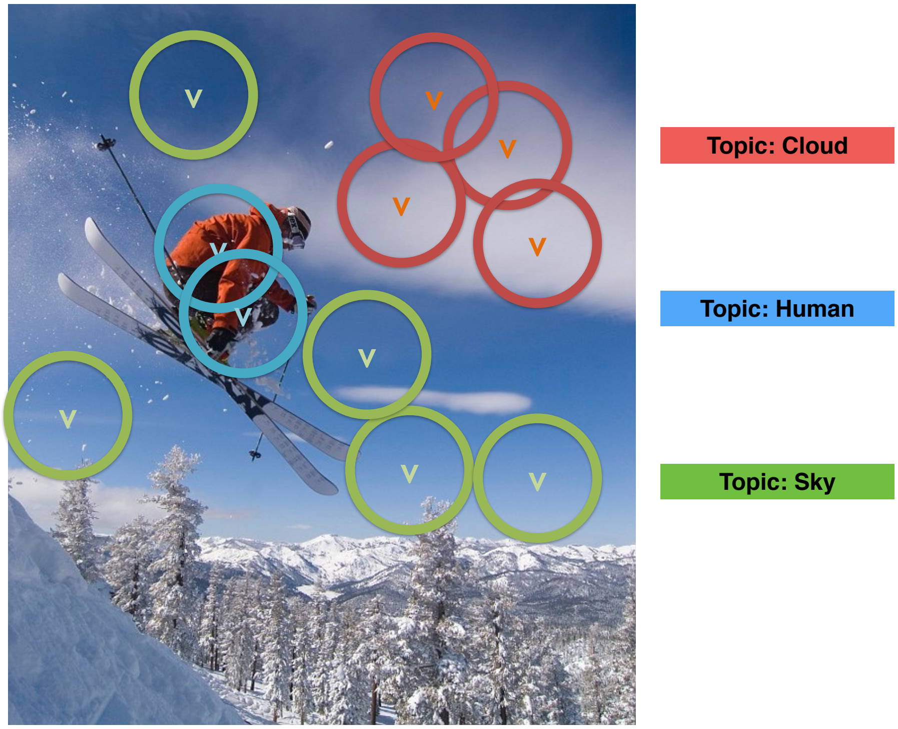


Figure 1.2: How dow we relate Topic models to Image processing? A pictorial explanation.

insight can be used to define basic facial patterns similar to manually encoded facial action units.

In this thesis we have studied topic models' usefulness in predicting continuous emotions from the point of view of dimension reduction techniques. The reason behind this is that what topic models do is to project image/audio features to a new topic simplex that is more informative and useful. We will now discuss some of the common dimension reduction techniques and discuss them in the light of topic models.

1.3 Dimension Reduction Techniques

While building an emotion recognition system, be it for discrete or continuous emotion predictions, we need to reduce the dimensions of the feature space given that the video and audio contain large number of features. A dimension reduction technique plays an important role in label prediction and in this thesis we have applied topic models to emotion prediction as dimension reduction techniques. Throughout this work we have evaluated topic models against many dimension reduction techniques and one of the most popular among them is the Principal Component Analysis. PCA model operates on a continuous feature space and projects data into a new feature space where the basis vectors are orthogonal to each other thereby reducing the redundancy. LDA can be viewed as the counterpart of PCA for discrete features. The basis vectors of the PCA are conceptually equivalent to the topics of LDA but the difference is that the topics can be visualized and also interpreted. While PCA models the linear dependencies within features, Kernel PCA [50] is a non linear extension to it. in KPCA the kernel trick is used where in the projections of the features are made from a higher dimensions space than the original feature space. It guarantees a linear projection when features are projected from a higher dimensional transformed space than the original space. KPCA thus gives a non linear perspective to dimension reduction techniques.

1.4 Extensions to Graphical structure of topic models

In recent years, probabilistic topic models have made significant contribution to both feature learning as well as supervised learning. From a set of visual/audio features extracted from video and audio frames, latent factors (topics) are learnt and used to predict emotions as well as changes in states. This latent topic space is richer than the original feature space as it considers correlations and co-occurrences of features. In the context of text and web mining several extensions to the basic graphical structure of topic models has been made to handle the application at hand. E.g. Hu et al [29] extended LDA to include dependencies between two different sets of text corpora like twitter feeds and event commentaries. Chang et al. [16] extended LDA to model links between documents thereby building a document relation network which is very useful in web mining. Extensions to topic models have also been proposed in some of the image processing applications, e.g. Caro et al. [14] have extended LDA to achieve better image segmentation using spatial regions within the model framework.

Since topic models are tuned to text documents, we need to quantize real-valued features. Topic models assume that data or words in a document are exchangeable which is not true for video data which has a temporal and spatial dependency in its features. Topic models have also not been explored in the context of emotion analysis and specifically for supervised concept drift problems. In this work all these challenges are touched upon and new topic models and quantization techniques to handle them are proposed.

In this research, apart from validating the LDA model to other dimension reduction techniques for emotion analysis, we have proposed several extensions. To model the continuous features without quantizing video and audio features, we have proposed Dirichlet Gaussian mixture model. To use the response variable in the process of dimension reduction we have used Supervised LDA [12] and validated them against supervised dimension reduction techniques like supervised PCA (SPCA) [9]. To avoid overfitting of supervised LDA models we have added regularization to the SLDA framework and evaluated their performance. Since multiple modalities like both audio and video contain different information about the emotions, it is more enriching to build models that consider them together. To achieve this we have proposed multi modal topic models that generate supervised topic by modeling audio and video features together rather than individually.

Emotion change detection is another important application where we have to predict only changes in emotions rather than the actual emotions. In order to predict change, we need to build models that have a temporal aspect to them. For this reason we have proposed two temporal extensions to topic models Adaptive Temporal Topic Model [35] and Supervised LDA for change detection [36], where we have incorporated dependency between pairs of consecutive documents into the topic models.

1.5 Overview

Chapter 2 discusses existing research in human affect analysis and probabilistic topic models and argues the significance and contributions of this research. Chapter 3 introduces the overall methodology used to address two applications 1) emotion recognition, and 2) emotion change detection (or concept drift). It also contains details about experimental methodology that describes how topic models have been applied to these specific applications. The different datasets, features used in this work and the evaluation techniques are also explained in Chapter 3.

The major contributions of this research are in area of probabilistic topic models and we have developed and experimented with four different categories viz. 1) unsupervised models, 2) supervised models, 3) continuous models, and 4) temporal models. Unsupervised topic models are discussed in Chapter 4 where the graphical models do not model the response variables. Chapter 5 contains discussion on supervised topic models where the response variable is also included in the model training. In addition, we also discuss multimodal fusion where different modalities are considered to predict responses. Chapters 4 and 5 contain evaluation and results on emotion recognition. Unlike emotion recognition, change detection needs a temporal component in the model so that the temporal changes in features can be mapped to changes in emotions. Chapter 6 delves into topic models which incorporate the time component into the graphical structure. In chapter 6 we have specifically looked at the emotion change detection problem and have evaluated topic features. Apart from our theoretical contributions, we have contributed to the Social Interaction Assistant project which has been discussed in Chapter 7.

Chapter 2

RELATED WORK

Automated detection and analysis of human behavior is one of the most popular research areas of this age due to its applicability to a wide range of applications like smart home technologies, pervasive healthcare, athletic training, security and surveillance etc . Human behavior analysis can be partitioned into two broader categories, 1) human activity/ gesture analysis, and 2) human affect/emotion analysis. The former deals with physical activities and the latter indicates the metal state of a person. The affective state of a human being can be perceived through vocal communication, facial movements, and certain body gestures. In this thesis we will be concerned with automatic recognition of human affective states using vocal communication and facial movements. This area of research is interdisciplinary in nature which has attracted attention of researchers from computer vision, machine learning, psychology and social sciences. We will address this problem from a computer vision and machine learning perspective. From this perspective human emotion recognition is yet another supervised learning problem.

2.1 Emotion Recognition Models

Face detection and tracking algorithms have achieved high amount of automation and sophistication in the area of computer vision. Whereas emotion recognition systems have not yet reached such level of accuracies especially in real time environments. And this is true for both discrete as well as continuous emotions. The primary areas of improvement in this area are the extraction of features that can explain emotions better and building of pattern recognition models specific to this application.

Two popularly used facial features are the shape/geometric and appearance based features. Shape features by themselves do not have enough information to explain subtle emotions and thus most of the papers use appearance descriptors instead. Facial landmarks extracted using active shape models (ASM) and active appearance models (AAM) are used as geometric features to predict emotions in [40], [8]. Mean appearance models , linear binary patterns (LBP) [52], local phase quantizations (LPQs) [61], histogram of gradients (HOG) [22], scale invariant feature transform (SIFT) [55], Gabor features [59] are few examples of appearance features used for emotion recognition.

Features for emotion analysis can be also categorized as spatial and temporal where the former retain the spatial dependencies between regions of images while the latter retain he temporal dependencies between audio or video frames. Some of the examples of spatial features are the Landmarks features, region based bag-of word features and facial fiducials based SIFT features. The most prominent temporal features are the Low Level Descriptor and Mell-Gibbs spectral coefficient audio features where these features are extracted using a window of frames. Gabor wavelet, Discrete Cosine Transforms and Fourier transforms are also few examples of temporal features. Some of the temporal features that are gaining popularity are the temporal bag of words features like LBPTop [41], LGBPTop [60] features which are extension of LBP [47] and LGBP [67] features to 3 dimensions.

Since features extracted from images can be correlated, noisy and redundant, selecting the best of them is common technique. Feature selection can be done using the labels and annotations (emotions) and is a supervised approach. AdaBoost is one of the most popular feature selection approaches and has been used for emotion recognition by Hao et al [57]. Another approach for supervised feature selection is cascade of classifiers approach as in Li et al [38] where SVM cascades are used to extract features. Since these models heavily rely on the labeled training data for validation, they tend to overfit the data. Once features are extracted, either their dimensionality is reduced using Principal Component Analysis [46], Independent Component Analysis [54] or a sparse representation is created using Singular Value Decomposition [68].

Audio Visual Emotion Challenge (AVEC) is one of the most popular emotion recognition challenges in recent years and the only competitions that deal with continuous emotion recognition systems. AVEC 2012 [52] through 2014 [60] have released datasets and features with annotated arousal and valence dimensions. These challenges have attracted quite few researchers to work on continuous emotion recognition. These challenges were able to produce benchmarks and state of the art research for continuous emotion recognition.

2.2 Topic models for Image and Video analysis

In general, a direct mapping of the features to emotions is avoided and instead, the feature space is projected to a lower dimensional space which is then mapped to the emotion space. Serious thought is not given to find a low dimensional space that retains as much information from the original space, considers correlations between features and ignores redundant and contradictory information. In text mining this problem is tackled using latent space models and topic models where the word features are projected to a latent topic space which is then used in unsupervised and supervised settings. Probabilistic latent semantic indexing (PLSI) model proposed by Hofmann [27] is a latent space model that gave a probabilistic dimension to latent semantic indexing that projects the word frequencies to a latent topic eigen space. Blei et al [13] have proposed Latent Dirichlet Allocation (LDA) model which is a more generalized topic model with explicit modeling of all the variables using multinomial-dirichlet conjugate distributions. In recent years topic models have been gaining popularity in image and video analysis, e.g. Li et al [24] used a broad estension of LDA to identify the events and objects in scene images, supervised LDA has been used for image annotation [17] and temporal topic models that were tailor made for video scene analysis [63].

There has not been much work on application of topic models in the affect analysis community. There are only two cited works that have used topic models for facial expression (discrete emotion) recognition. Temporal latent topic model (TLTM) proposed by Shang & Chang [34] is an extension to LDA that considers each image as a document and uses the temporal dependency between adjacent images to predict emotion from a image sequence. Shang [53] developed a discriminative model that uses an asymmetric dirichlet prior and weights the image words to predict emotions. Both of these works have focused on six basic discrete emotions and considered facial landmarks and datasets that contain artificially stimulated expressions. In this work we propose to apply topic models to complex continuous emotions and real valued appearance based image descriptors. Along with unsupervised topic models we also have used supervised topic models so that the emotions can affect the topics that are extracted.

As time passes, the distribution of both features and labels can change and in many applications predicting these changes is useful. Especially, predicting changes in emotions is useful in medical diagnosis, federal investigations, human computer interaction and many other applications. There has been some work in detecting changes in emotion from human speech but not from facial videos and images. And surprisingly there have been not much work on topic models for concept drift based applications making it interesting to see how topic models can be applied to emotion change detection which is proposed in next chapters.

Chapter 3

EXPERIMENTAL METHODOLOGY

This chapter provides an overview of the experimental methodology that has been followed throughout this thesis. The major contributions of this thesis are int the area of dimension reduction and supervised learning and the thesis has been structured to highlight the performance of probabilistic topic models. Figure 3.1 contains a snapshot of all the techniques we have evaluated and developed. The models annotated with an asterix have been contributed through this research and we have used the rest of the models for prototyping and benchmarking.

We have evaluated topic models' performance as dimension reduction techniques in the context of Emotion recognition and change detection applications. We have used five different features in this work which have been discussed in detail in Section 3.1. Since topic models need quantized features, we performed feature quantization on all the features and more details can be found in Section 3.1. Once the features are quantized, we applied different probabilistic topic models depending on the application at hand. We have used Latent Dirichlet Allocation (LDA), Supervised LDA (SLDA) and Regularized SLDA (RSLDA) on quantized features for dimension reduction to predict emotions and emotion changes. We used Dirichlet Gaussian Mixture model (DGMM) on continuous features and in the context of emotion recognition only. Adaptive Temporal Topic Model (ATTM) and Supervised LDA for Change detection (SLDACD) have been developed and evaluated to predict emotion changes only. Each of these models will be discussed in detail in Chapters 4, 5 and 6 and have been compared with prominent dimension reduction techniques like Principal Component Analysis (PCA), Kernel PCA and Supervised CPA (SPCA).

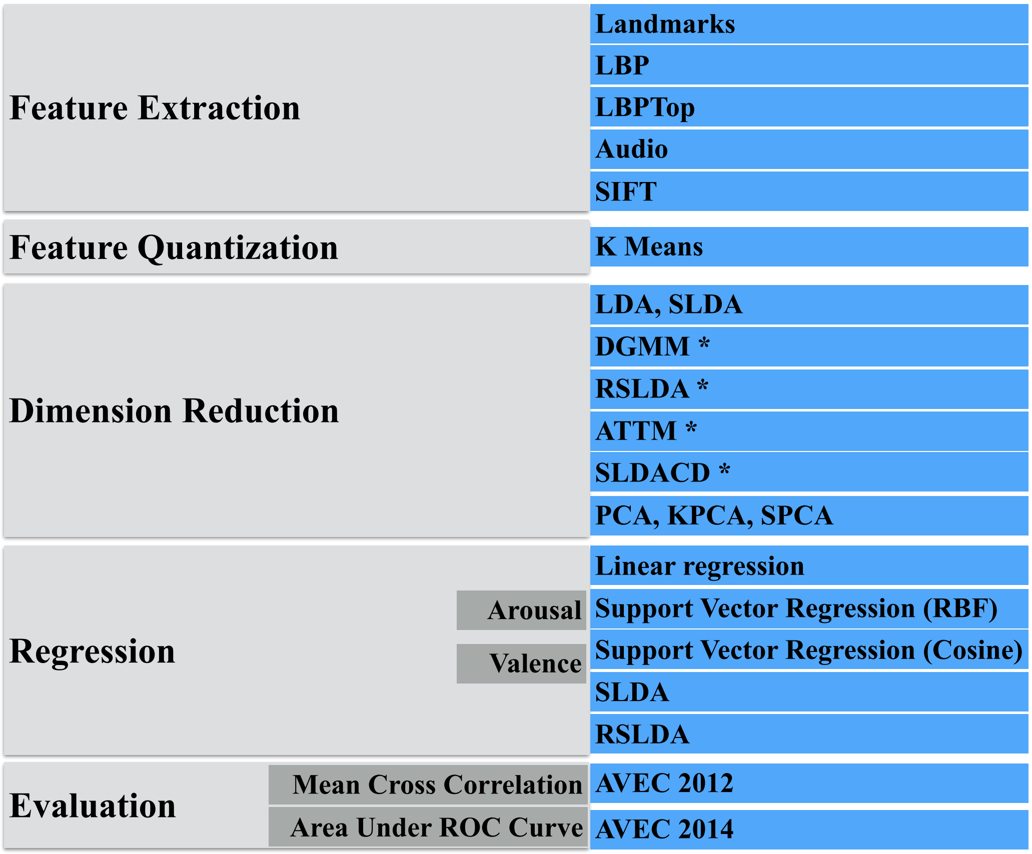


Figure 3.1: A snapshot of all the algorithms and features explored in this thesis

The performance of the reduced feature spaces has been tested on two continuous emotions *arousal* and *valence* using different regression models like, Linear regression, Support Vector regression with RBF and Cosine kernels, and supervised topic models like SLDA and RSLDA. In order to evaluate the performance we have used two publicly available continuous emotion datasets AVEC2012 and AVEC2014 that contain annotations of arousal and valence. These datasets will be explored in Section 3.2 where we also define the evaluation criteria used throughout this thesis, mean cross

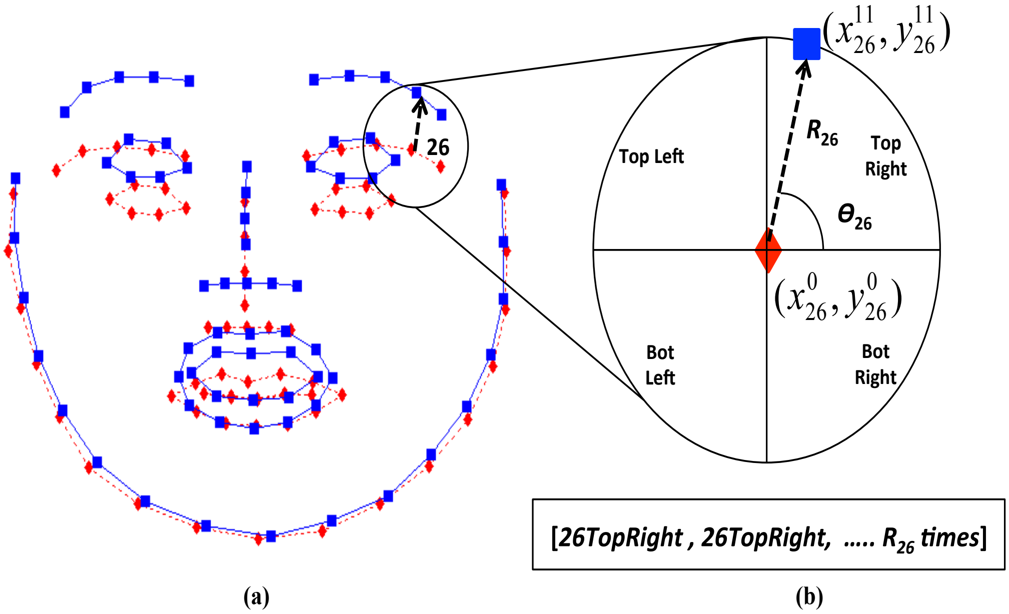

26
$(x_{26}^{11}, y_{26}^{11})$
R_{26}
Top Left
Top Right
θ_{26}
(x_{26}^{0}, y_{26}^{0})
Bot Left
Bot Right
[26TopRight , 26TopRight, ... R_{26} times]
(a)
(b)

Figure 3.2: Landmark features are quantized using Neutral and Current frames

correlation and area under ROC curves for emotion recognition and change detection applications.

3.1 Feature Extraction and Quantization

In our experimentations we have used both video and audio frames for emotion recognition, and to use topic models, the real valued features need to be quantized to words. In this work we have used some of the popularly used video and audio features as base features and quantized them into words. Below are the description of all features that have been used in this work.

Landmark Features

49 facial landmarks are extracted from the neutral and peak frames of a video. We then discretize the relative movements of these points in terms of two variables, r, the amount of movement, and θ, the direction of movement. The r values are obtained through a histogram of K bins, which are generated uniquely for each landmark L_i, by considering the movements of that landmark L_i across all available face image sequences. The θ values are quantized into 4 bins - top left, top right, bottom left, and bottom right. Hence, given the neutral and peak frames of the video, a *facial document* for the video is generated using θ_is for each of the landmarks $L_i, i = 1, \ldots, 49$, as the individual *facial words*, and then repeating these facial words r_i times in the document thus capturing the amount or intensity of movement. Figure 3.2(a) illustrates how the facial words for a sample landmark point are generated. These facial words represent the temporal movement of landmarks and their counts represent the intensity of movement. We have used the software provided by Intraface API [66] to detect, align faces and extract the facial landmark points. Even though the quantization is meaningful, there is a caveat that we always need a neutral frame for this. We have tackled this issue by defining the first frame of a video sequence as the *neutral frame* . The logic behind this is that the emotional state of a person is a relative to the user's state of mind for that particular session.

Linear Binary Pattern (LBP) features

Linear Binary Pattern features were introduced by Ojala [47] for texture analysis but have been shown to be promising for emotion analysis by Jiang et al. [30] in their work on sparse representations for expression analysis. Local Binary Patterns (LBPs) that are calculated at each pixel are used to generate facial words. An image is divided

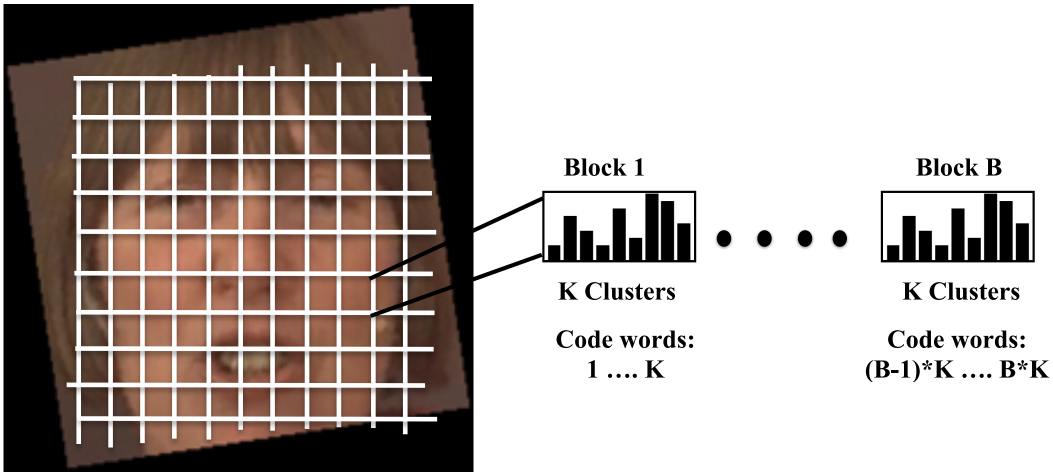


Figure 3.3: Linear Binary Pattern features are quantized using K Means clustering on each block

into smaller image blocks in the XY direction and the histogram of the counts of 59 unique uniform LBPs is calculated for each block. Depending on the number of image blocks B, the total dimensionality of the features turns out to be $B \times 59$. The histograms from all images that belong to the block B are considered and quantized to K clusters using K-Means algorithm. Since these vectors are multidimensional, we need to define a distance metric for the K-Means algorithm. In general histograms are normalized vectors or frequency counts, and in either way cosine distance is a popular distance measure for such vectors. So we have used K-Means algorithm with cosine distance to cluster each of the B blocks and the total number of code words that are generated are $B \times K$. The formula for calculating cosine distance between two vectors is given below:

$$Cosine(\boldsymbol{x},\boldsymbol{y}) = 1 - \frac{\sum\limits_{n} x_n y_n}{\sqrt{\sum_n x_n^2}\sqrt{\sum_n y_n^2}}$$

After clustering the features, each 59-dimensional histogram belonging to block b are assigned the one of clusters say k and their code would be $(b-1)*K+k$ where K is the total number of clusters per block. Figure 3.3 demonstrates pictorially the quantization procedure for LBP features. Since the same quantization algorithm has been used for all the video features, it is useful to provide an algorithmic form to it as in in Algorithmm 1.

Algorithm 1 Algorithm to vector quantize real values features

```
Input: Feature vector $\boldsymbol{X}$ , # of clusters $K$, # of blocks $B$ , $distance\_function$
Output: Quantized Feature vector $\tilde{\boldsymbol{X}}$
procedure VectorQuantize
    for each block $b$ do
        Cluster the feature vectors $\boldsymbol{X_b}$ using K-Means clustering with $distance\_function$
    end for
    for each sample $n$ do
        for each block $b$ do
            Assign the cluster id of the block $nb$ as the new quantized feature $\tilde{X}_{nb}$
        end for
    end for
end procedure
```

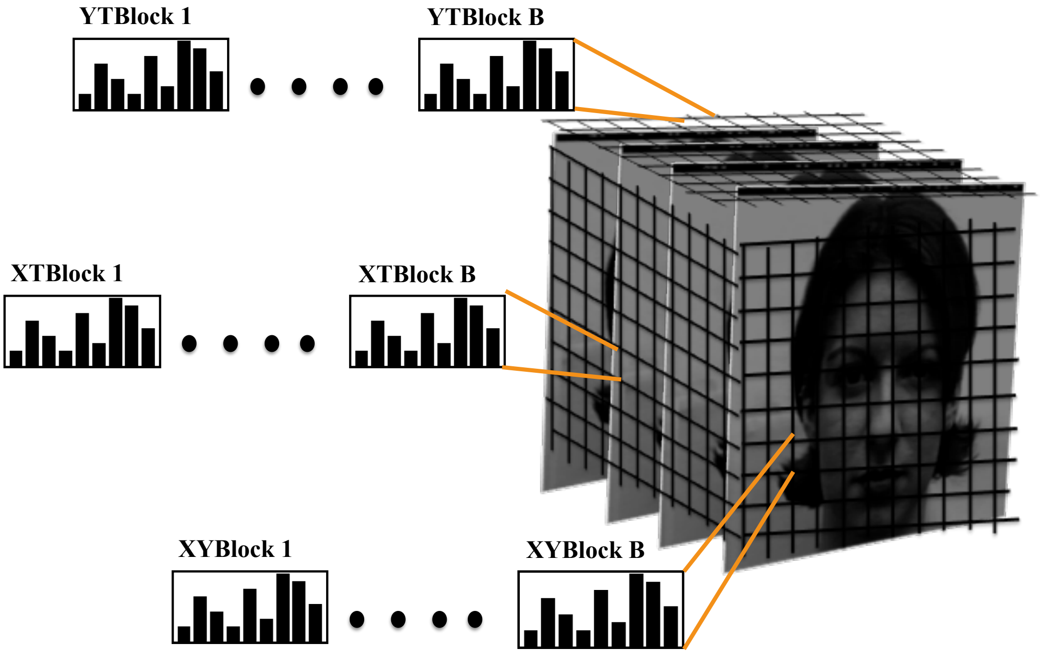


Figure 3.4: LBP Top features are quantized performing K Means clustering on each block in each of the XY, XT and YT dimensions

LBP Top features

LBPTop features are an extension to LBP features and have been proposed in [41] specific to the context of video analysis. At each time step the consecutive frames that succeed the current frame including it are considered and for each of the B blocks in the XY dimension voxels of size $X_{bx}Y_{by}T$ are considered, where X_b and Y_b are the dimensions of block b. As shown in Figure 3.4, we obtain three histograms for each block each from their respective dimension and these histograms are appended as $1x177$ dimensional features and so the total size of the LBPTop feature vector will be $B * 177$. Following a similar quantization technique as in LBP, we cluster each of the $3 * B$ block histograms using K Means clustering. The code word for a b-th histogram is assigned as $(b-1) * K + k$ where $b = 1 \dots 177 * B$ and K is the number of clusters. We have extracted LBPTop features by detecting the faces using Viola Jones [64] algorithm and have used the API provided by Zhao et al. [41] to extract LBPTop features from facial block voxels.

LGBP features

Linear Gabor Binary Pattern features have been proposed in [67] as an extension to LBP features. To extract LGBP features, each video frame is convolved using 18 different Gabor wavelet filters proposed by Lee et al. [37]. LBP features are then extracted from each of the B blocks of 18 of the transformed images and appended to form a vector of $B*18*59$ dimensions. Since the feature vector is still a concatenation of 59-dimensional histograms we used the same quantization technique as in LBP and LBPTop features with cosine distance as the distance function for clustering.

SIFT features

Scale Invariant Feature Transforms (SIFT) are among the most popular features proposed by Lowe [39] in which images are transformed with Difference of gaussians at different scales. In general the key points are automatically selected by SIFT technique but since we are aware that the key points for facial movements are the fiducial points, we instead extracted the SIFT features at 49 landmarks and 22 interpolated points between these points as shown in Figure 3.5. We used a fixed scale of 2 to extract the features and concatenated the vectors from different orientations at each point. We have used euclidean distance to quantize the vectors using Algorithm 1 where the number of blocks $B = 71$. We have used the Matlab based SIFT toolbox provided by [5].

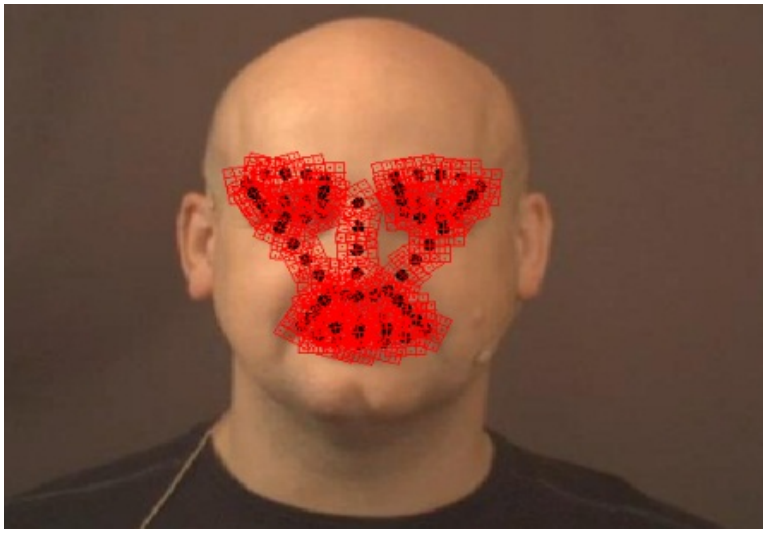


Figure 3.5: Plot of SIFT [39] features extracted from 71 interpolated facial fiducial points

Audio Features

Following the approach taken by [49], 25 energy and spectral LLDs each with 42 functionals and 6 voice related LLD with 32 functionals each have been selected making it a total of 1242 features. In order to apply topic models each audio feature is clustered into K clusters using K-Means algorithm and then each feature in every audio document is replaced with its cluster-id.

3.2 Evaluation

Since this work deals primarily with continuous emotion recognition, we have worked on datasets released under the Audio Visual Emotion Challenges (AVEC) organized by the Social Signal Processing Network (SSPNET) [3] which is a European agency to enhance human emotion and behavior sensing and analysis. AVEC challenges have been organized since 2011 through 2014 at major conference venues like ACM International Conference on Multimodal Interaction (ICMI), ACM International conference on Multimedia (ACMMM), IEEE International Conference on Multimedia and Expo (ICME). As part of these challenges three major datasets have been released each year and the most prominent have been the AVEC2012 [52], AVEC2013 [61] and AVEC2014 [60] datasets. Each of these have released raw videos and text of conversations, challenge-specific audio and video features, continuous emotion labels for arousal and valence. In all of these datasets the videos contain recorded sessions of facial videos of users having a conversation with Audiovisual sensitive Artificial Listener (ASAL) that engage users in discussions varying across the range of emotions. Few examples of the ASAL's engaging the users are shown in Figure 3.6.

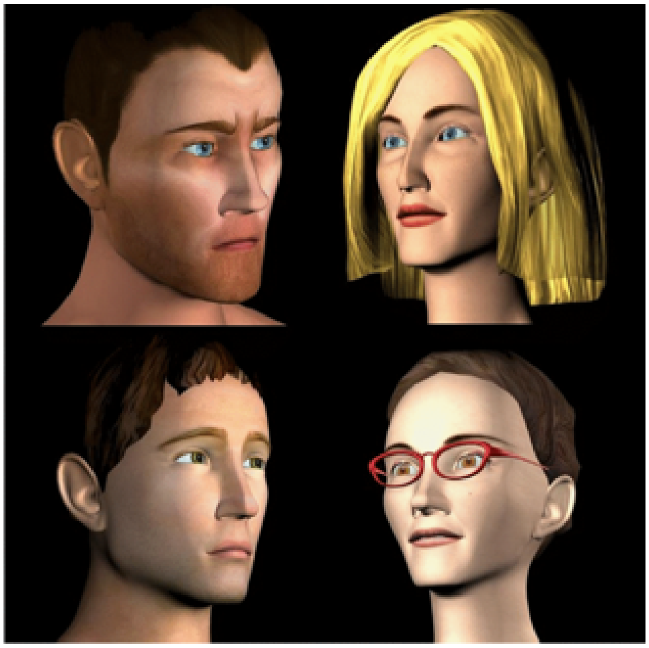


***Photo Courtesy*: IEEE © 2009, Valstar et al,**
"A demonstration of audiovisual sensitive artificial listeners"

Figure 3.6: An example from [51] of how Sensitive Artificial Listeners look like.

In each of the above datasets, each facial video is annotated with the continuous emotions arousal and valence at each frame. Thus each video is associated with two

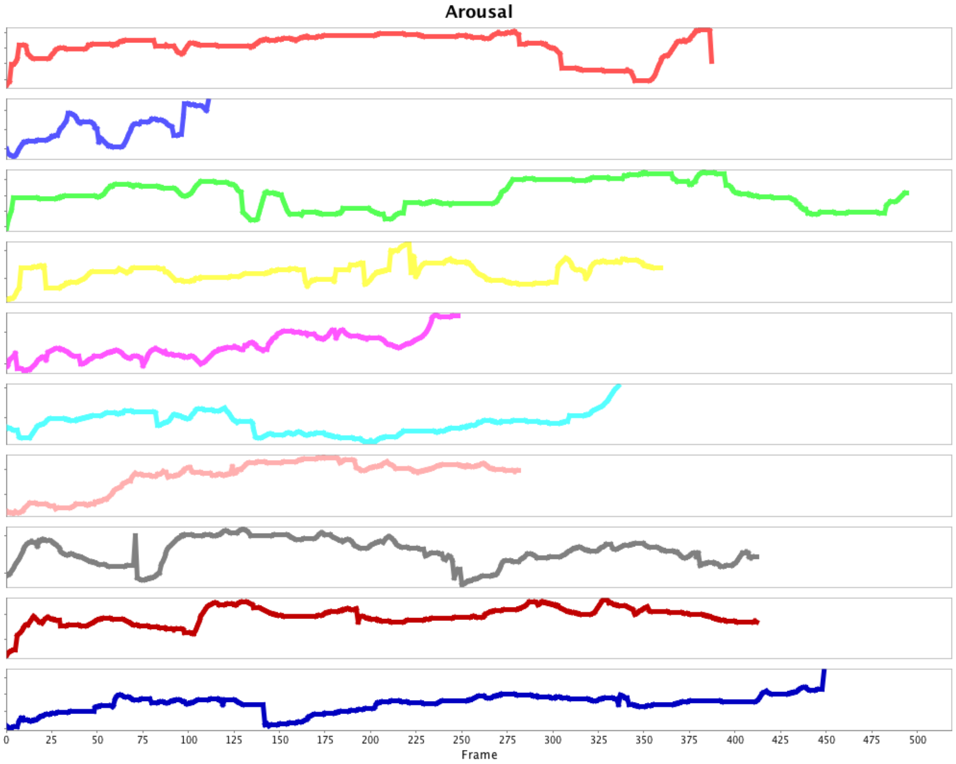


Figure 3.7: Plots of Arousal for sample videos from AVEC 2012 [52]

emotion based time series. Figure 3.7 shows sample plots of the arousal dimension for the first 10 training videos of the AVEC2012 dataset. The AVEC datasets contain a set facial videos (includes audio) divided into Training, Development and Test sets. The labels for training and development sets are available whereas the test labels are not available for testing the accuracies. While building and testing the models, the training data is used to select different parameters by performing cross validation which are then used to make predictions on development set. And similarly while generating predictions for the test sets both training and development sets. Throughout this work training and testing has been done on training and development videos.

Once the continuous emotions are predicted, the performance of the models is evaluated using mean cross correlation and area under ROC curves for recognition and change detection. The Mean Pearson Cross-correlation is given by:

$$\text{MeanCorrelation} = \left| \frac{1}{V} \sum_{v} \frac{\sum\limits_{n} (y_{vn} - \bar{\boldsymbol{y}}_v)(\tilde{y}_{vn} - \bar{\tilde{\boldsymbol{y}}}_v)}{\sqrt{\sum\limits_{n} (y_{vn} - \bar{\boldsymbol{y}}_v)^2} \sqrt{\sum\limits_{n} (\tilde{y_{vn}} - \bar{\tilde{\boldsymbol{y}}}_v)^2}} \right| \tag{3.1}$$

where V is the total number of videos, $\boldsymbol{y}$ and $\tilde{\boldsymbol{y}}$ are the actual and predicted emotion time series values, $\bar{\boldsymbol{y}}$ and $\bar{\tilde{\boldsymbol{y}}}$ are the mean values of the emotion time series. Thus higher the correlation over more videos, the better would be the model's predictive capabilities. The ROC (Receiver Operating Characteristic) curves are popularly used to evaluate classification or change detection models. Change detection is a binary classification problem and the model predictions are used to calculate True Positive Rate (TPR) and False Positive Rate (FPR) given by:

$$TPR = \frac{TP}{TP + FN}$$
$$FPR = \frac{FP}{FP + TN}$$

where TP, FN, FP and TN indicate the true positive, false negative, false positive and true negative predictions from the classifier. To plot ROC curves, the predictions from the model are extracted using different threshold or confidence parameters and the (FPR, TPR) coordinates are plotted. The area under the ROC curves is calculated across the models and the one with better area indicates better prediction capabilities.

In Chapter 4 we will begin our description of topic models and inference and then evaluate their performance in comparison to other models.

Chapter 4

UNSUPERVISED TOPIC MODELS

In Chapter 2 we have discussed the utility of topic models in mining feature co-occurrences and also their contribution to predictive analytics. Topic models have been traditionally used to categorize text and web corpora into meaningful grouping so that their retrieval becomes faster and accurate. They have also been used in a supervised learning setting to predict movie or review ratings. The most popular unsupervised topic model is Latent Dirichlet Allocation (LDA) proposed by Blei et al. [13] which is a probabilistic generative model. In Section 4.1 of this chapter we will focus on probabilistic topic models that are unsupervised by introducing LDA and the inference techniques as they lay the foundation for the rest of the models throughout this thesis. We will interpret the topics extracted using LDA across features and evaluate their performance on continuous emotion recognition. Since LDA model operates on quantized features we discuss a new Dirichlet Gaussian Mixture Model (DGMM) in Section 4.2 as an extension of LDA to continuous features.

Let $\boldsymbol{X}$ be an m x n matrix of document samples, where each sample represents a feature vector of frequencies of words occurring in the sample. Latent Semantic Indexing (LSI) model proposed by Hoffman [27], performs a Singular Value Decomposition on this matrix to obtain document-to-topic ($\boldsymbol{D}$) , topic-topic ($\boldsymbol{S}$) and topic-term ($\boldsymbol{T}$) matrices as shown in the Figure 4.1.

The $\boldsymbol{D}$ matrix can be considered as a new feature matrix and these features are a projection of the original term vector space to the topic vector space. Unlike prevalent projection techniques, LSI also gives an interpretation of the new topic features, where $\boldsymbol{T}$ represents topic definitions. A probabilistic interpretation to LSI

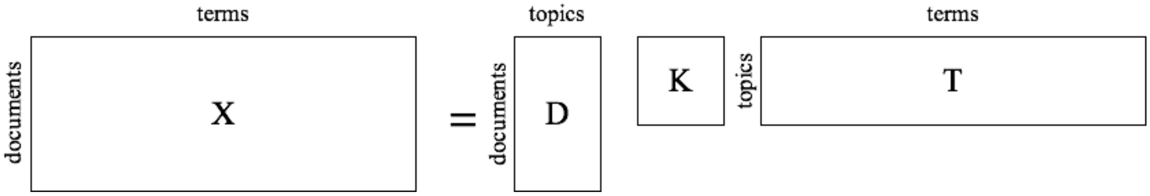
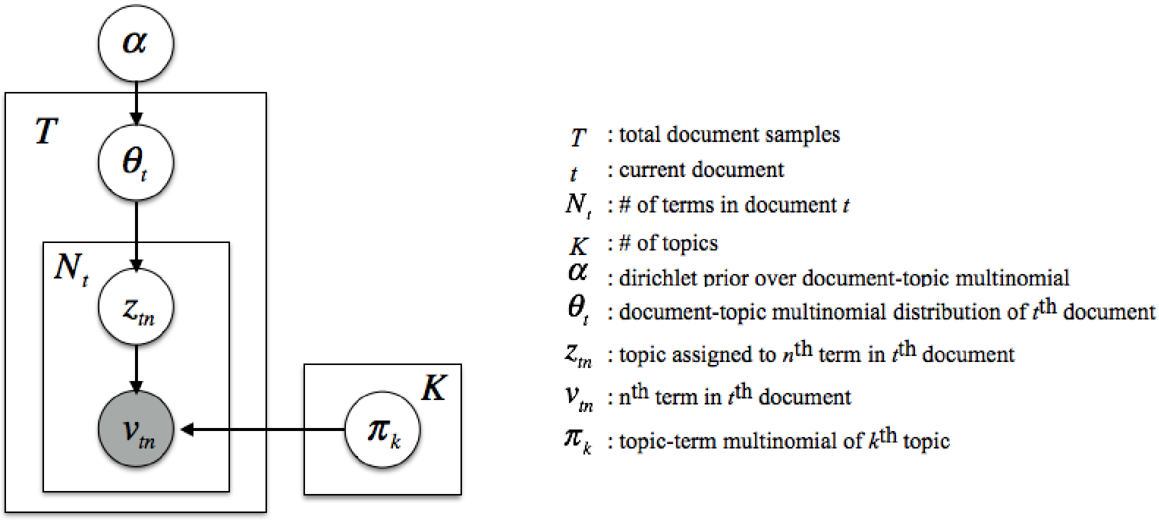


Figure 4.1: Singular Value Decomposition of document-term matrix into document-topic and topic-term matrices

was proposed by Hoffman [27] as pLSI model where each document sample is modeled using a Multinomial distribution. pLSI model is a double mixture model and fits a mixture of Multinomials instead of a mixture of Gaussian as in Gaussian Mixture Models. Another advantage of having a probabilistic interpretation is that it can be used as a generative model to generate new documents given the model. Blei [13] have highlighted that even though pLSI model makes the inference more meaningful, the model does not perform well on unseen documents especially when they contain new co-occurrences of words.

Latent Dirichlet Allocation proposed by Blei [13] address the pitfalls of pLSI model by make it more generalizable. LDA model is a double mixture model with priors attached to the mixture distribution itself. LDA assumes a Dirichlet prior over the mixture of topics for each document sample, and the assumption is made because Dirichlet distribution is the conjugate for Multinomial distribution. Similar to pLSI, LDA is also a generative model where given the set of Dirichlet parameters, it can generate document samples. This chapter deals with different unsupervised topic models used in this work and since LDA is the basis for both unsupervised and supervised models, it will be discussed in much detail.



T : total document samples
t : current document
N_t : # of terms in document t
K : # of topics
α : dirichlet prior over document-topic multinomial
θ_t : document-topic multinomial distribution of tth document
z_{tn} : topic assigned to nth term in tth document
v_{tn} : nth term in tth document
π_k : topic-term multinomial of kth topic

Figure 4.2: Graphical model for Latent Dirichlet Allocation (LDA) model

4.1 Latent Dirichlet Allocation model

Figure 4.2 shows the graphical model for LDA model with the description of random variables in the inset. $\boldsymbol{\theta_t}$ is a Multinomial distribution of topics in a document t and $\boldsymbol{\phi_k}$ is a Multinomial distribution of words in a topic k. Given a set of K topics, each word v_{tn} in a document t is generated by assigning a topic z_{tn} from the distribution $\boldsymbol{\theta_t}$ and then generating a word from the topic distribution $\boldsymbol{\pi_{z_{tn}}}$. Now, given a set of documents, we need to infer the distributions $\boldsymbol{\theta_t}$ and $\boldsymbol{\pi_k}$. To make LDA a better generative model, Dirichlet prior distributions are assumed over each of the Multinomial distributions $\boldsymbol{\theta_t}$ and $\boldsymbol{\pi_k}$. $\boldsymbol{\alpha}$ are the Dirichlet prior parameters for $\boldsymbol{\theta_t}$. Even though a Dirichlet prior can be assumed over the distribution $\boldsymbol{\pi_k}$ to make the Multinomial distributions much smoother, we will ignore this parameter in the following discussion.

A Dirichlet distribution is a multivariate extension to the 2-dimensional Beta distribution and in considered as a Distribution over distributions. A Dirichlet distribution takes an K-dimensional probability vector $\boldsymbol{x}$ where $\sum_k x_k = 1$ and is parameterized by an K-dimensional vector $\boldsymbol{\alpha}$ where $\alpha_k > 0$. The probability density function for a Dirichlet distribution is given by [2]:

$$\mathrm{Dir}(\boldsymbol{x}, \boldsymbol{\alpha}) = \frac{1}{B(\boldsymbol{\alpha})} \prod_{k=1}^{K} x_k^{\alpha_i - 1}$$

where the Beta function B is given by $B(\boldsymbol{x}) = \frac{\prod_{i=1}^{K} \Gamma(x(i))}{\Gamma(\sum_{i=1}^{K}(x(i)))}$ and Γ is a gamma function. The Dirichlet distribution assigns probabilities to probability distributions there by giving a generic shape to the distribution space instead of assuming a particular distribution. The $\boldsymbol{\alpha}$ parameters influence the shape of the Dirichlet distribution, where a symmetric $\boldsymbol{\alpha}$ implies uniform importance to all K dimensions and an unsymmetric $\boldsymbol{\alpha}$ implies the other way. For $\alpha > 1$, the shape is concave whereas for $\alpha < 1$ the shape becomes convex (boat shaped). For a Multinomial distribution, a smooth estimation of its parameters can be obtained using a Dirichlet prior over its parameters. Since Dirichlet distribution is a conjugate prior, the posterior distribution of Multinomial parameters is also a Dirichlet distribution, which eases the Bayesian inference of these parameters.

LDA model is generative model and thus its structure can be used to generate data samples. While creating generative graphical models it is extremely important to show they can be used to generate samples. Below is the generative process of Latent Dirichlet Allocation model with K topics.

Generative Process of LDA:

1. Draw Multinomial K-topic distribution $\boldsymbol{\theta} \sim \mathrm{Dirichlet}(\boldsymbol{\alpha})$

2. For each of the N words,

   (a) Assign a topic, $z_n \sim \mathrm{Multinomial}(\boldsymbol{\theta})$ to n^{th} word

   (b) Draw a term $w_n \sim \mathrm{Multinomial}(\boldsymbol{\pi}_{z_n})$

In LDA model the only observed variable is $\boldsymbol{v}$ whereas $\boldsymbol{\theta_t}$ and $\boldsymbol{z}_t$ are latent variables. and the parameters to be estimated are the Dirichlet priors $\boldsymbol{\alpha}$ and the Multinomials $\boldsymbol{\pi_k}$. The likelihood of observing a document $\boldsymbol{v_t}$ given these parameters is:

$$p(\boldsymbol{v_t}|\boldsymbol{\alpha},\boldsymbol{\pi}) = \int_{\boldsymbol{\theta_t}} p(\boldsymbol{\theta_t}|\alpha) \sum_{\boldsymbol{z_t}} \prod_{n=1}^{N_t} p(v_{tn}|\boldsymbol{\pi}_{z_{tn}}) p(z_{tn}|\boldsymbol{\theta}_t) \boldsymbol{d\theta_t} \tag{4.1}$$

The parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\pi}$ can be estimated by maximizing the logarithm of the likelihood function given by Eq 4.1. But the term $p(v_{tn}|\boldsymbol{\pi}_k)p(k|\boldsymbol{\theta}_t)$ inside the integral couples the variables $\boldsymbol{\theta}$ and $\boldsymbol{\pi}$ thereby making this calculation intractable during estimation. Due to the availability of latent variables $\boldsymbol{\theta}$ and $\boldsymbol{z}$, we can estimate the parameters using the Expectation Maximization (EM) algorithm [21] where the complete likelihood $p(\boldsymbol{v}, \boldsymbol{z}, pmb\theta)$ is maximized instead of the incomplete likelihood $p(\boldsymbol{v})$. Since various graphical models and their parameter estimation techniques will be discussed throughout this work, it is important to understand how EM algorithm works.

4.1.1 EM Algorithm for LDA

In situations where it is difficult to optimize the incomplete likelihood function, EM algorithm can be used instead to optimize the complete likelihood of observed $\boldsymbol{v}$ and hidden variables $\boldsymbol{z}$. . Let $q(\boldsymbol{z})$ be a distribution over $\boldsymbol{z}$ and for any given $q(\boldsymbol{z})$ and consider the following derivation where $\boldsymbol{\beta}$ represents all the parameters to be estimated and $\boldsymbol{Z}$ represents all latent variables in the model.

$$
\begin{aligned}
\log(p(\boldsymbol{v}|\boldsymbol{\beta})) = & \sum_{\boldsymbol{Z}} q(\boldsymbol{Z}) \log(p(\boldsymbol{v}|\boldsymbol{\beta}))(\because \text{constant} = E[\text{constant}]) \\
= & \sum_{\boldsymbol{Z}} q(\boldsymbol{Z}) \log(p(\boldsymbol{v}|\boldsymbol{\beta})) + \sum_{\boldsymbol{Z}} q(\boldsymbol{Z}) \log(p(\boldsymbol{Z}|\boldsymbol{v},\boldsymbol{\beta})) - \sum_{\boldsymbol{Z}} q(\boldsymbol{Z}) \log(p(\boldsymbol{Z}|\boldsymbol{v},\boldsymbol{\beta})) \\
= & \sum_{\boldsymbol{Z}} q(\boldsymbol{Z}) \log(p(\boldsymbol{v},\boldsymbol{Z}|\boldsymbol{\beta})) - \sum_{\boldsymbol{Z}} q(\boldsymbol{Z}) \log(p(\boldsymbol{Z}|\boldsymbol{v},\boldsymbol{\beta})) \\
= & \sum_{\boldsymbol{Z}} q(\boldsymbol{Z}) \log(p(\boldsymbol{v},\boldsymbol{Z}|\boldsymbol{\beta})) - \sum_{\boldsymbol{Z}} q(\boldsymbol{Z}) \log(q(\boldsymbol{Z})) \\
& + \sum_{\boldsymbol{Z}} q(\boldsymbol{Z}) \log(q(\boldsymbol{Z})) - \sum_{\boldsymbol{Z}} q(\boldsymbol{Z}) \log(p(\boldsymbol{Z}|\boldsymbol{v},\boldsymbol{\beta})) \\
= & \underbrace{\sum_{\boldsymbol{Z}} q(\boldsymbol{Z}) \log \frac{p(\boldsymbol{v},\boldsymbol{Z}|\boldsymbol{\beta})}{q(\boldsymbol{Z})}}_{\mathcal{L}(q,\boldsymbol{\beta})} + \underbrace{\sum_{\boldsymbol{Z}} q(\boldsymbol{Z}) \log \frac{q(\boldsymbol{Z})}{p(\boldsymbol{Z}|\boldsymbol{v},\boldsymbol{\beta})}}_{\text{KLD}(q(\boldsymbol{Z}),p(\boldsymbol{Z}|\boldsymbol{v},\boldsymbol{\beta}))} \qquad (4.2)
\end{aligned}
$$

In the above derivation KLD denotes the Kullback-Leibler Divergence [33] between q and the posterior distribution $p(\boldsymbol{Z}|\boldsymbol{v},\boldsymbol{\beta})$, which is a measure of dissimilarity between the distributions. Since by definition KLD $\geqslant 0$, the above derivation implies that for any distribution q over the latent variables, the first term $\mathcal{L}(q,\boldsymbol{\beta})$ is always less than or equal to log likelihood. So the first term is the lower bound for likelihood and through EM algorithm we try to maximize the lower bound to indirectly increase the log likelihood.

Figure 4.3 gives a visual interpretation to the Expectation Maximization algorithm. The algorithm begins with a random assignment of values to the parameters and in each iteration i, the posterior $p(\boldsymbol{Z}|\boldsymbol{v},\boldsymbol{\beta}_i)$ is calculated using current parameters and the distribution q in the formula for $\mathcal{L}(q,\boldsymbol{\beta})$ is replaced with the posterior to obtain LB_i. The lower bound LB_i is then maximized by estimating the parameters $\boldsymbol{\beta}_{i+1}$ which will be used in the next iteration. The convergence of the algorithm is indicated by the amount of change in the log likelihood function values between iterations. Given the above interpretation, EM algorithm thus contains two steps, E-step where the lower bound is calculated as an expectation of the posterior distribution and an M-step where the calculated lower bound is maximized with respect to the

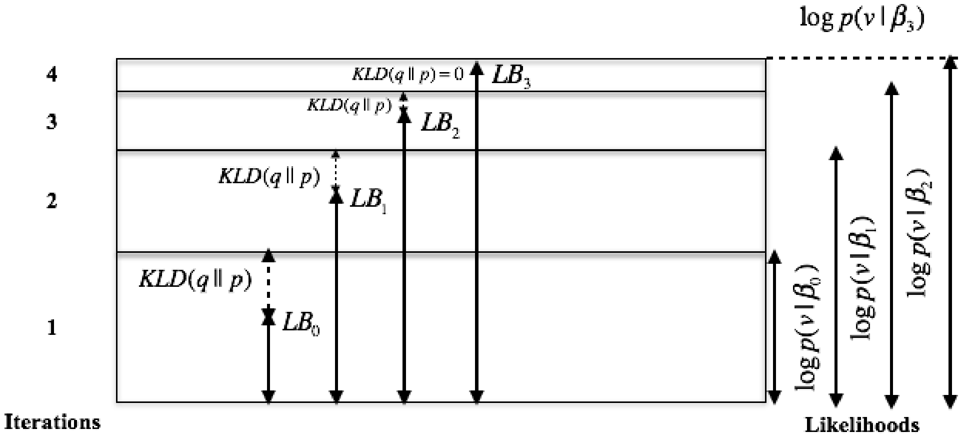


Figure 4.3: Interpretation of Expectation Maximization Algorithm. After each iteration, the difference between the lower bound (LB) and log likelihood diminishes.

model parameters thereby estimating the optimal parameters. Algorithm 2 demonstrates the EM algorithm using the context and variables defined in Latent Dirichlet Allocation model..

Since EM algorithm requires evaluation of the Posterior distribution, for LDA it is given by:

$$p(\boldsymbol{\theta}, \boldsymbol{z} | \boldsymbol{v}, \boldsymbol{\alpha}, \boldsymbol{\pi}) = \frac{p(\boldsymbol{\theta}, \boldsymbol{z}, \boldsymbol{v}, \boldsymbol{\alpha}, \boldsymbol{\pi})}{p(\boldsymbol{v} | \boldsymbol{\alpha}, \boldsymbol{\pi})} \tag{4.3}$$

It can be observed that the above posterior is also intractable as the denominator contains the likelihood term which is difficult to calculate due to the above mentioned couplings. This makes it difficult to estimate LDA parameters directly using the plane vanilla EM algorithm shown in Algorithm 2. In these circumstances, two popular methodologies are used viz. Gibbs Sampling [26] or Variational Expectation Maximization [31]. Gibbs sampling, as the name suggests is a sampling technique used to estimate the parameters of an intractable distribution where random variables are sampled iteratively. During each iteration independent variables are sampled one at a times assuming the rest to be constant and continuing until convergence. The values

Algorithm 2 Expectation Maximization Algorithm

```
Input: All documents $\boldsymbol{v}$
Latent Variables: Topic assignment vectors $\boldsymbol{z}$ and document-topic distributions $\boldsymbol{\theta}$
Parameters: Dirichlet Priors $\boldsymbol{\alpha}_0$ and Multinomial topic-term distributions $\boldsymbol{\pi}_0$
Initialize the parameters to random values e.g. assign equal probabilities to all terms in $\boldsymbol{\pi}$
$i := 0$
procedure EM ALGORITHM($I$)                              ▷ $I$ is # of iterations
    while $i < I$ or !$converged$ do                     ▷ Iterate till convergence
        E-Step:
        (i) Calculate the Posterior $p(\boldsymbol{\theta}, \boldsymbol{z}|\boldsymbol{v}, \boldsymbol{\alpha}_{i-1}, \boldsymbol{\pi}_{i-1})$ using Eq 4.3
        (ii) Derive the lower bound $\mathcal{L}(q, \boldsymbol{\alpha}_{i-1}, \boldsymbol{\pi}_{i-1})$ using the above Posterior
        M-Step:
        (i) Estimate the parameters $\boldsymbol{\alpha}_i, \boldsymbol{\pi}_i$ that maximize $\mathcal{L}(q, \boldsymbol{\alpha}_{i-1}, \boldsymbol{\pi}_{i-1})$
        $i := i + 1$
    end while
end procedure
```

sampled so far are used during the current iteration and convergence is guaranteed after a certain time period called the Burn-in period. The samples extracted during the burn-in period are used to estimate the parameters of the distribution. In the following section, we will delve into Gibbs sampling equations for LDA model.

4.1.2 Gibbs Sampling Algorithm for LDA

For LDA, Gibbs sampling involves sampling the topic of each word z_{tn} conditioned on topics assigned to the rest of the words. Among the two methodologies discussed above, even though Gibbs sampling techniques can assure globally optimal solution, deriving the equations is not straight forward. In this context Collapsed Gibbs sampling is a popular method to derive sampling equations and Griffiths [56] provided the derivations for LDA model. By definition Collapsed sampling techniques sample the latent distribution $\boldsymbol{z}$ by integrating over (collapsing) the variables $\boldsymbol{\theta}$ and $\boldsymbol{\pi}$. In the following equations a Dirichlet prior β is assumed on the topic-term distributions $\boldsymbol{\pi}$ as well. The posterior distribution of $\boldsymbol{z}$ is given by:

$$\begin{aligned} p(\boldsymbol{z}|\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{v}) &\propto p(\boldsymbol{v}|\boldsymbol{z},\boldsymbol{\beta})p(\boldsymbol{z}|\boldsymbol{\alpha}) \\ &= \int p(\boldsymbol{v}|\boldsymbol{z},\boldsymbol{\pi})p(\boldsymbol{\pi}|\boldsymbol{\beta})\boldsymbol{d\pi} \int p(\boldsymbol{z}|\boldsymbol{\phi})p(\boldsymbol{\theta}|\boldsymbol{\alpha})\boldsymbol{d\theta} \\ &= \prod_t \frac{B(\boldsymbol{\alpha}+\boldsymbol{n}_t)}{B(\boldsymbol{\alpha})} \prod_k \frac{B(\boldsymbol{\beta}+\boldsymbol{n}_k)}{B(\boldsymbol{\beta})} \end{aligned}$$

where B is the the Beta function. For a given word with index (t,n), the topic is sampled using this sampling equation:

$$\begin{aligned} p(z_{tn}=k|\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{v},\boldsymbol{z}_{\neg(tn)}) &= \frac{p(\boldsymbol{v},\boldsymbol{z}|\boldsymbol{\alpha},\boldsymbol{\beta})}{p(\boldsymbol{v},\boldsymbol{z}_{\neg(tn)}|\boldsymbol{\alpha},\boldsymbol{\beta})} && (4.4) \\ &\propto \frac{B(\boldsymbol{n}_t+\boldsymbol{\alpha})}{B(\boldsymbol{n}^t_{-(t,n)}+\boldsymbol{\alpha})} \frac{B(\boldsymbol{n}_k+\boldsymbol{\beta})}{B(\boldsymbol{n}^k_{-(t,n)}+\boldsymbol{\alpha})} && (4.5) \end{aligned}$$

where $\boldsymbol{n_t}$ is the vector of counts of topics assigned to document t, $\boldsymbol{n_k}$ is the vector of counts of terms assigned to topic k and $\neg(tn)$ implies all but the index (tn). Algorithm 3 explains the overall procedure for Gibbs sampling and variable estimation. For a given number of iterations I, pre-assigned parameter values $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ and input documents $\boldsymbol{v}$ the procedure iterates till burn in period. Step 3 samples the topics for each word using Eq 4.5 and once all topics are assigned in current iteration, the

Algorithm 3 Collapsed Gibbs sampling algorithm for LDA by [56]

```
Input: All documents, $\boldsymbol{v}$                         ▷ Step 1
Initialize $\boldsymbol{z}$ to topics 1 to K using Uniform(1,K)     ▷ Step 2
procedure GIBBS($I$)                                ▷ $I$ is # of iterations
    for iter = 1 to $I$ do                          ▷ Iterate till burn-in
        for $t = 1$ to $T$ do                       ▷ Iterate over documents
            for $n = 1$ to $N_t$ do                 ▷ Iterate over words
                Sample $z_{tn}$ using $p(z_{tn}|\boldsymbol{z}_{\neg tn}, \boldsymbol{v})$   ▷ Step 3
            end for
        end for
        Update counts $\boldsymbol{n_k}$ and $\boldsymbol{n_t}$              ▷ Step 4
    end for
    Update the parameters $\boldsymbol{\theta}$ and $\boldsymbol{\pi}$          ▷ Step 5
end procedure
```

counts are updated in Step 4. And in Step 5 the values of $\boldsymbol{\theta}$ and $\boldsymbol{\pi}$ are updated using the expectations of the following distributions:

$$p(\boldsymbol{\pi}_k|\boldsymbol{v}, \boldsymbol{\beta}) \approx Dirichlet(\boldsymbol{\pi}_k; (\boldsymbol{\beta} + \boldsymbol{n}_k)) \tag{4.6}$$

$$p(\boldsymbol{\theta}_t|\boldsymbol{v}, \boldsymbol{\alpha}) \approx Dirichlet(\boldsymbol{\theta}_t; (\boldsymbol{\alpha} + \boldsymbol{n}_t)) \tag{4.7}$$

A stark distinction between Gibbs Sampling and EM algorithm is that the parameters like $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are assumed to be known in the former methodology whereas the latter has a way to estimate these parameters. Unlike Gibbs sampling which is a probabilistic approach to estimate parameters, Variational EM algorithm is a deterministic way of estimation built upon the structure of EM algorithm. The following section explains how variational EM approach is used to estimate the parameters of LDA.

4.1.3 Variational EM Algorithm for LDA

Variational EM, given by Jordan et al. [31] considers alternatives to posterior distribution which is evaluated in the EM algorithm. As we observe in Figure 4.3, in each step we aim to reduce the KL divergence between the distribution q and the posterior. If the posterior can be evaluated, posterior itself can be selected as the distribution q, but in cases otherwise, different q distributions are considered and the optimal distribution that is most closest to the actual posterior is selected. The distribution q is called a surrogate as it is trying to replace or simulate the actual posterior and the method is called Variational EM because it derives concepts from the Variational Calculus. Since the complex relationships between random variables makes a posterior intractable, this methodology tries to simplify the graphical model by dropping some edges and models the new posterior distributions indexed by variational parameters. This relaxation also helps in choosing a simpler family of distributions for optimization.

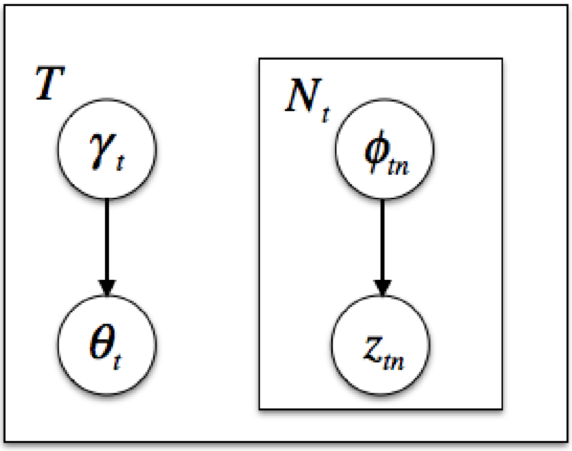


Figure 4.4: A simplified graphical model for LDA to approximate the actual posterior using simple surrogate family of distributions.

Figure 4.4 shows a simpler graphical model for LDA where the coupling between $\boldsymbol{\theta}$ and $\boldsymbol{\pi}$ have been simplified by removing the edges between $\boldsymbol{\theta}$ and $\boldsymbol{z}$ and removing the observed variables $\boldsymbol{v}$ from the model. The joint distribution of this model represented as q is given by:

$$q(\boldsymbol{\theta}_t, \boldsymbol{z}_t | \boldsymbol{\gamma}_t, \boldsymbol{\phi}_t) = q(\boldsymbol{\theta}_t | \boldsymbol{\gamma}_t) \prod_{n=1}^{N_t} q(z_{tn} | \boldsymbol{\phi}_{tn})$$

Thus the new distribution is defined over the latent variables of the original model, $\boldsymbol{z}$ and $\boldsymbol{\theta}$ but with new parameters $\boldsymbol{\gamma}$ and $\boldsymbol{\phi}$. In the simple version, the observed variables $\boldsymbol{v}$ are ignored and thus the variational parameters are in a way dependent on $\boldsymbol{v}$. To find the optimal variational parameters, the Kullback Leibler Divergence between q and posterior p is minimized which is given below:

$$[\boldsymbol{\gamma}^*(\boldsymbol{v}), \boldsymbol{\phi}^*(\boldsymbol{v})] = \underbrace{\text{arg min}}_{\boldsymbol{\gamma}, \boldsymbol{\phi}} \log \Big(q(\boldsymbol{\theta}, \boldsymbol{z} | \boldsymbol{\gamma}, \boldsymbol{\phi}) \frac{q(\boldsymbol{\theta}, \boldsymbol{z} | \boldsymbol{\gamma}, \boldsymbol{\phi})}{p(\boldsymbol{\theta}, \boldsymbol{z} | \boldsymbol{v}, \boldsymbol{\alpha}, \boldsymbol{\pi})} \Big)$$

As shown in the EM algorithm, minimizing the KL-divergence is equivalent to maximizing the lower bound $\mathcal{L}(\boldsymbol{\gamma}, \boldsymbol{\phi}; \boldsymbol{\alpha}, \boldsymbol{\pi})$. Expanding this lower bound using Eq 4.2 we have:

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\gamma}, \boldsymbol{\phi}; \boldsymbol{\alpha}, \boldsymbol{\pi}) &= E_q \Big[\log \Big(\frac{p(\boldsymbol{v}, \boldsymbol{\theta}, \boldsymbol{z} |, \boldsymbol{\alpha}, \boldsymbol{\pi})}{q(\boldsymbol{\theta}, \boldsymbol{z} | \boldsymbol{\gamma}, \boldsymbol{\phi})} \Big) \Big] \\
&= E_q[\log(p(\boldsymbol{\theta}|\boldsymbol{\alpha})) + \log(p(\boldsymbol{z}|\boldsymbol{\theta})) + \log(p(\boldsymbol{v}|\boldsymbol{z}, \boldsymbol{\pi})) - \log(q(\boldsymbol{\theta}, \boldsymbol{z}|\boldsymbol{\gamma}, \boldsymbol{\phi}))] \\
&= E_q[\log(p(\boldsymbol{\theta}|\boldsymbol{\alpha}))] + E_q[\log(p(\boldsymbol{z}|\boldsymbol{\theta}))] + E_q[\log(p(\boldsymbol{v}|\boldsymbol{z}, \boldsymbol{\pi}))] \\
&\quad - E_q[q(\boldsymbol{\theta})] - E_q[q(\boldsymbol{z})] \qquad (4.8)
\end{aligned}$$

Expanding each of the terms in Eq 4.8 we have:

$$\begin{aligned}
E_q[\log(p(\boldsymbol{\theta}|\boldsymbol{\alpha}))] &= \log\left(\frac{\Gamma(\sum\limits_k \alpha_k)}{\prod\limits_k \Gamma(\alpha_i)}\right) + E_q[\log(\prod_k \theta_k^{\alpha_k - 1})] \\
&\quad \text{(used definition of Dirichlet distribution)} \\
&= \log(\Gamma(\sum_k \alpha_k)) - \sum_k \log(\Gamma(\alpha_i)) + E_q[\sum_k (\alpha_k - 1)\log(\theta_k)] \\
&= \log(\Gamma(\sum_k \alpha_k)) - \sum_k \log(\Gamma(\alpha_i)) + \sum_k (\alpha_k - 1)(\Psi(\gamma_k) - \Psi(\sum \gamma_k)) \\
&\quad \text{(where } \Psi \text{ is a Digamma function derived a in [13])}
\end{aligned}$$

$$\begin{aligned}
E_q[\log(p(\boldsymbol{z}|\boldsymbol{\theta}))] &= E_q[\log(\prod_n \prod_k \theta_n^{z_{nk}})] \\
&\quad \text{(where} z_{nk} = 1 \text{ if } z_n = k, z_{nk} = 0 \text{ otherwise)} \\
&= \sum_n \sum_k E_q[z_{nk}] E_q[\log(\theta_n)] \\
&= \sum_n \sum_k \phi_{nk}(\Psi(\gamma_k) - \Psi(\sum \gamma_k))
\end{aligned}$$

$$\begin{aligned}
E_q[\log(p(\boldsymbol{v}|\boldsymbol{z},\boldsymbol{\pi}))] &= \sum_n \sum_k q(z_{nk}) \log(\pi_{z_{nk},v_n}) \\
&= \sum_n c_n \sum_k \phi_{nk} \log(\pi_{k,v_n}) \\
&\quad \text{(where } c_n = \text{count of word } v_n\text{)}
\end{aligned}$$

$$\begin{aligned}
E_q[q(\boldsymbol{\theta})] &= \log(\Gamma(\sum_k \gamma_k)) - \sum_k \log(\Gamma(\gamma_i)) + \sum_k (\gamma_k - 1)(\Psi(\gamma_k) - \Psi(\sum \gamma_k)) \\
&\quad \text{(used definition of Dirichlet distribution)}
\end{aligned}$$

$$\begin{aligned}
E_q[q(\boldsymbol{z})] &= \sum_n \sum_k q(z_{nk}) \log(q(z_{nk}) \\
&= \sum_n \sum_k q(z_{nk}) \log(q(z_{nk}) \\
&= \sum_n \sum_k \phi_{nk} \log(\phi_{nk})
\end{aligned}$$

Appendix B.1 contains Java code that calculates the above mentioned lower bound which indeed is the log likelihood of the given data.

Maximizing $\mathcal{L}$ with respect to $\boldsymbol{\phi}$ and $\boldsymbol{\gamma}$ involves constrained optimization. To solve for the Multinomials $\boldsymbol{\phi}$, we solve the following optimization function that deals with terms in $\mathcal{L}$ that contain only $\boldsymbol{\phi}$:

$$\begin{gathered}\text{maximize } \phi_{nk}(\Psi(\gamma_k) - \Psi(\textstyle\sum \gamma_k)) + \phi_{nk}\log(\pi_{k,v_n}) - \phi_{nk}\log(\phi_{nk}) \\ \text{subject to } \sum_k \phi_{nk} = 1\end{gathered}$$

Converting this optimization to include the constraint using the Lagrange Multipliers [11], the maximization function is transformed to:

$$\text{maximize } \phi_{nk}(\Psi(\gamma_k)-\Psi(\sum \gamma_k))+\phi_{nk}\log(\pi_{k,v_n})-\phi_{nk}\log(\phi_{nk})+\lambda_n(\sum_k \phi_{nk}-1) \quad (4.9)$$

Differentiating Eq 4.9 with respect to Multinomial ϕ_{nk} and equating it to zero we obtain the optimal value as given below:

$$\phi_{nk} \propto \pi_{kn}\exp(\Psi(\gamma_k) - \Psi(\sum \gamma_k)) \quad (4.10)$$

Differentiating Eq 4.9 with respect to the Dirichlet parameter γ_k and equating it to zero, we obtain the optimal value as given below:

$$\gamma_k = \alpha_k + \sum_n \phi_{nk} \quad (4.11)$$

The above formula indicates intuitively that the Dirichlet prior parameter for topic k is the sum of the probabilities of all words being assigned to topic k. We can observe that estimating γ requires prior estimates of ϕ, so each of these parameters are estimated iteratively until convergence. Maximizing the lower bound with respect to variational parameters is the E-step of the variational EM algorithm. Once these

Algorithm 4 Variational Bayes algorithm for LDA by [13]

```
Input: All documents $\boldsymbol{v}$ and Dirichlet Priors $\boldsymbol{\alpha}$
Parameters: Multinomial topic-term distributions $\boldsymbol{\pi}_0$
Output: Document-Topic distributions $\boldsymbol{\gamma}$ and Topic-Term distributions $\boldsymbol{\pi}$
Initialize the parameters to random values e.g. assign equal probabilities to all terms in $\boldsymbol{\pi}_0$
$i := 0$
procedure LDAVBAlgorithm($I$,$J$)    ▷ $I$ is # of EM iterations, $J$ is # of E iterations
    while $i < I$ or !$converged$ do    ▷ Iterate till convergence
        E-Step:
        for each document $t$ do
            Initialize equal values to all topics in $\boldsymbol{\gamma_t}$ as $\gamma_{tk} = \alpha_k + \frac{N_t}{K}$
            $j := 0$
            while $j < J$ or !$converged$ do
                (i) Calculate the Multinomials $\boldsymbol{\phi_t}$ using Eq 4.10
                (ii) Calculate the Dirichlet priors $\boldsymbol{\gamma_t}$ using Eq 4.11
                $j := j + 1$
            end while
        end for
        M-Step:
        (i) Estimate $\boldsymbol{\pi}_i$ using Eq 4.12
        $i := i + 1$
    end while
end procedure
```

parameters are estimated, as in the regular EM algorithm, the parameters $\boldsymbol{\pi}$ and $\boldsymbol{\alpha}$ will be estimated in the M-step. But throughout this work we have chosen to select $\boldsymbol{\alpha}$ values using cross validation rather than direct estimation because this work mostly deals with supervised learning where we have access to prediction accuracies. Maximizing the lower bound 4.8 with respect to the parameter $\boldsymbol{\pi}$ gives the estimate as:

$$\pi_{kw} = \sum_{t}\sum_{n} \phi_{tnk} I_{nw} \tag{4.12}$$

$$\text{where } I_{nw} = \begin{cases} 1 \text{ if } n = w \\ 0 \text{ if } n \neq w \end{cases}$$

The above formula is intuitive because at the end of the iteration it consolidates the topic assignments to all the words to estimate the topic-term distributions. Algorithm 4 contains the flow of estimation of parameters in LDA using Variational Bayes approach.

By analyzing the space and time complexities of the Variational EM algorithm the total space complexity needed to estimate the optimal topic assignments is of the order $O((T * N) + (K * V))$ where T, N, K, V are number of documents, average number of words per document, number of topics and total vocabulary size respectively. The total time complexity of the algorithm is $O(I * ((T * J * N * K) + (K * V)))$ where I is the total number of iterations and the first and second terms are for running the E and M steps respectively. This complexity increases drastically as the number of documents increase and to address this issue in this work we have used Map Reduce [19] methodology to make the training process parallelizable. This parallelization would not be possible in the implementation of Gibbs Algorithm 3 as the topic assignments in each document are dependent on rest of the documents. Whereas in Variational EM each document can be dealt separately and thus enabling us to train the LDA model in parallel. Appendix B.2 provides the java implementation of the Map reduce algorithm for Variational E-step. We can observe that the number of documents that can assigned topics in parallel depends on the available system processors. A Mapper is spawned per each document that performs the E-step as a child thread and yields to the parent thread. Once the output from all documents is received, the M-step is then executed using all the $\boldsymbol{\phi}$ values to estimate $\boldsymbol{\pi}$.

Inference on Unseen Documents:

Typical of any generative model, the process of estimation of parameters of a graphical model like LDA is called Learning. Once the topic-term distributions $\boldsymbol{\pi}$ are learnt we can use the same EM set up to estimate the topic assignments for a new unseen document. This is called inference and we use the E-step from Algorithm 4 to estimate the document-topic distributions of a new document. We do not need the M-step for inference as we do not want to alter the topic-term distributions. But in recent years researchers have proposed online algorithms for LDA where the topic-term distributions are updated as and when a new document is observed. In this work we restrict our focus to offline EM learning and assume that the topic definitions are unaltered once they are learnt during training. In fact we will address the learning process as LDA training and inference on new documents as testing. Since this work concerns with supervised learning we use the features extracted using LDA along with classifiers and regressors to predict either discrete or continuous response labels. It has to be noted that the topic distributions that are used as features are derived by normalizing the variational parameters $\boldsymbol{\gamma}$.

4.1.4 Experiments and Results

We hypothesize that 1) topic models are able to extract latent feature patterns co-occurring in a set of image sequences, 2) the latent topics extracted by topic models can be visualized and have a semantic interpretation and 3) latent topics evolve with emotions over time and thus are able to predict continuous affect dimensions. In Section 4.1.4, latent topics extracted from different features are visualized and analyzed and it also contains results from LDA model on emotion recognition. A comparative analysis of LDA's performance across different features is also provided but before diving into these results, we present the experimentation methodology and evaluation criteria.

Experimental Setup

Using different feature quantization techniques described in Chapter 3, we have generated one *document* per video frame. For our pilot studies we have considered the AVEC12 dataset [52] that contains 31 training videos and 32 development videos. We have evaluated all the models and features on continuous emotion prediction of *arousal* and *valence* dimensions. Since the predictions and actual labels are time series values, we have used mean Pearson cross correlation across all test videos as the evaluation criteria. Before we analyze the results we will discuss the feature parameters used for this study. In this evaluation we have considered Landmarks (LM) [66], SIFT [39], LBP [47], LBPTop [41], and MFCC-LLD (Audio) [45] features as base features from which latent topic features have been extracted. The tools used to align faces and extract relevant features have been explained in detail in Section 3.1.

Since each of these features have been quantized using K-means and since the number of clusters K is a parameter, we selected the K values using cross validation on the training videos. Table 4.1 contains the distance measures and the number of clusters that gave the best cross validation results over emotion recognition. The third column in Table 4.1 contains the final vocabulary size for each of these features and here is how we arrive at them, for LM features it is 49 landmarks x 4 directions, for

Base Feature	Distance Function	# of Clusters	Vocabulary
LM	Euclidean	20	196
LBP	Cosine	50	5000
LBPTop	Cosine	50	13500
SIFT	Euclidean	100	7100
Audio	Euclidean	50	62100

Table 4.1: The distance functions and number of clusters used to quantize each of the base features in K-Means algorithm and these parameters are specific to AVEC12 data.

LBP features it is 100 blocks x 50 clusters, for LBPTop it is 90 blocks x 3 dimensions x 50 clusters, for SIFT features it is 71 key points x 100 clusters and for audio features it is 1242 LLDs x 50 clusters. Please refer to Table 4.2 for more detailed cross validation results where we evaluated all parameters using 3-fold cross validation and using same set of LDA parameters across the board. We have used Linear Regression models and trained them using topic features. The optimal parameters that maximized the performance of LDA-Linear models have been selected. Some of the distance functions are not applicable to all features as the vector dimension of the vectors that are quantized is one.

Using the ideal parameters, we have generated the image and audio documents with quantized words and used LDA model to extract the latent topics. Since LDA model is an iterative technique dependent on the convergence of the likelihood values, we have plotted the log likelihood values calculated using Eq 4 after each EM iteration. Figure 4.5 contains plots of changes in log likelihood values for LDA model on LM and LBPTop features on AVEC2012 training videos. We observe that the convergence is

# of Clusters	Distance Function	LM	LBP	LBPTop	SIFT	Audio
20	Euclidean	**0.13**	0.17	0.14	0.15	0.18
	Cosine	NA	0.18	0.15	NA	NA
50	Euclidean	0.127	0.21	0.19	**0.18**	**0.21**
	Cosine	NA	**0.23**	**0.24**	NA	NA
100	Euclidean	0.11	0.14	0.18	0.13	0.20
	Cosine	NA	0.13	0.20	NA	NA

Table 4.2: 3 Fold Cross-validation results on Arousal prediction on AVEC12 training videos that are used to select Feature quantization parameters. LDA-Linear regression models have been used to select the optimal parameters.

slower for the LBPTop features in comparison to LM features due to the difference in vocabulary size where it takes longer time for the model to stabilize when the vocabulary is larger.

Interpreting Latent Topics

One of the primary motivations to use Topic models is that it is a double mixture model where one mixture (document-topic mixture) is used to predict response values whereas the second mixture (topic-term mixture) comes handy in visualizing facial patterns that are common across users. In fact we hypothesize that these patterns can be considered as the building blocks for different facial movements. Also, when topic models are perceived as dimension reduction techniques, unlike PCA where the projection may not have a visual interpretation (but for the fact that they are the eigen vectors of the covariance matrix), LDA's basis vectors or topics can be represented semantically. In this context topic models can also be addressed as *Semantic*

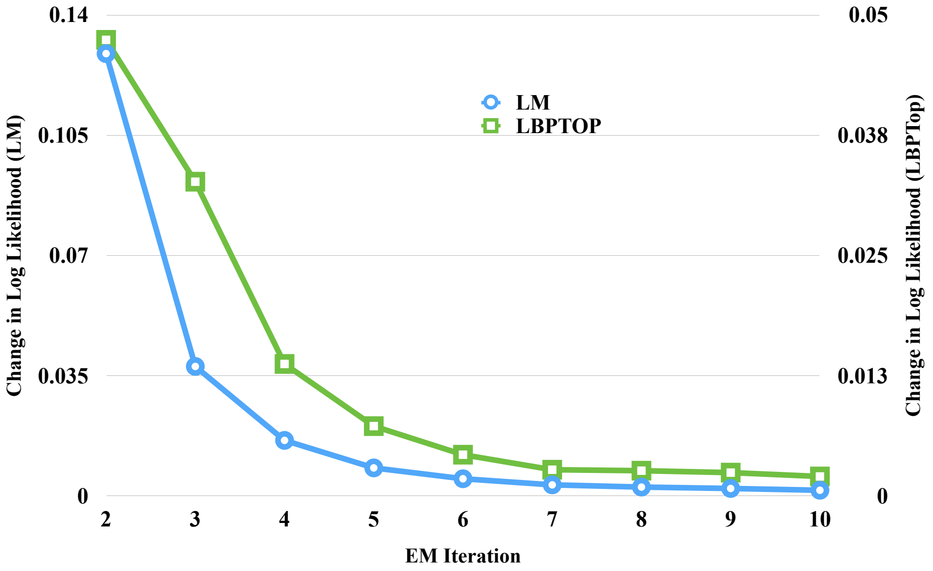


Figure 4.5: Changes in likelihood after each EM iteration in LDA model for Landmark and LBPTop features

Projection techniques. Since LDA model is built upon quantized features extracted from exiting audio, shape and geometric features which we call the base features, the topics extracted from each base feature has a different visual interpretation.

Figure 4.6 contains sample plots of LDA based topics obtained from SIFT features. As explained earlier, we have extracted sift features at 71 fiducial interpolated points as the key points in which case a term represents the quantized SIFT feature at a fiducial point. Thus a topic is defined a Multinomial over over quantized fiducial points where the higher the probability of a term implies that fiducial point has a higher affect over the topic. In this context Figure 4.6 contains plots of four different topics where the radius of the bubble represents the probability of that fiducial point

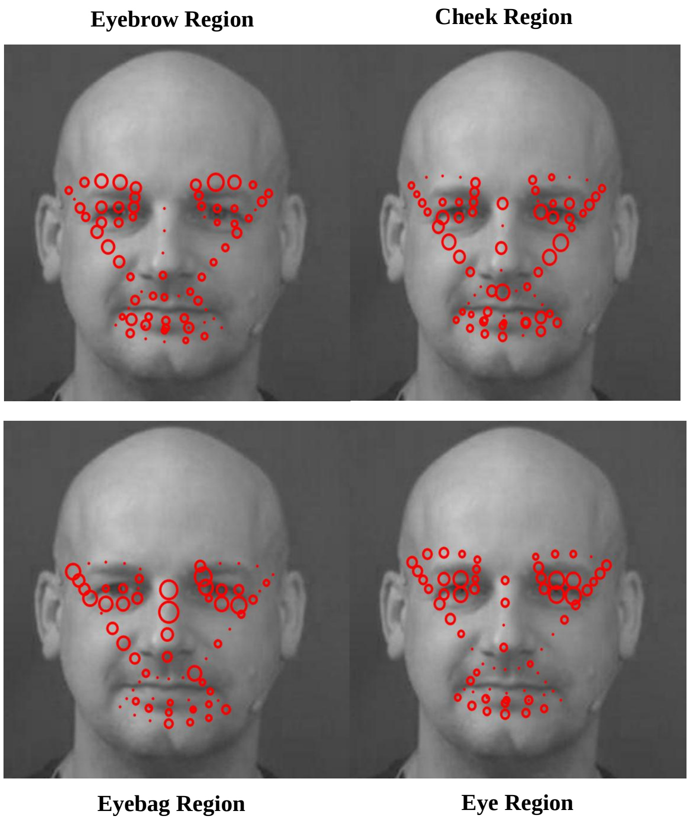


Figure 4.6: Sample plots of Topics extracted using LDA on SIFT features

in the topic. We can observe that each topic is influenced by a particular region of the face e.g. in the figure we see that the topics are influenced by eyebrow, cheek, eyeball and eye regions respectively.

One of the most prominent facial changes that we perceive in our daily lives are the facial landmark movements that are caused by the underlying facial muscles. We have used LDA on the landmark features where the angle and movement of each landmark is quantized to terms. In this case the topics are defined over direction of movement of landmarks e.g. *35-top-left* which signifies that this topic models the

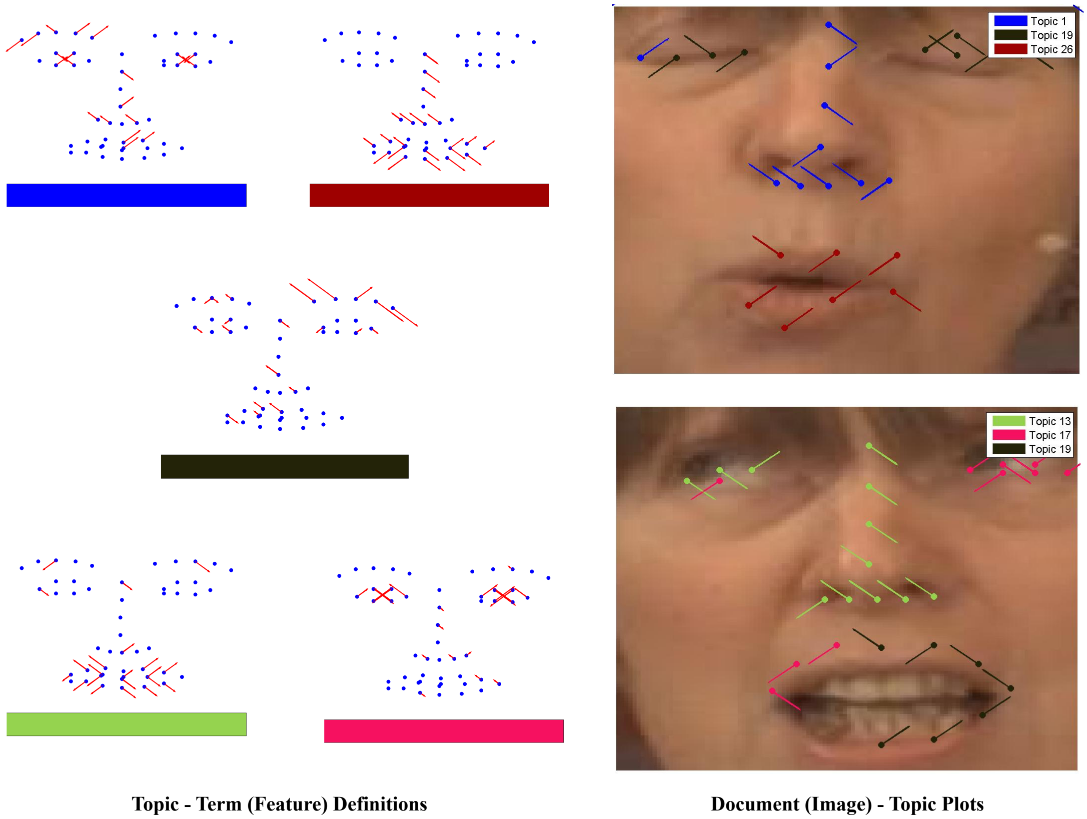
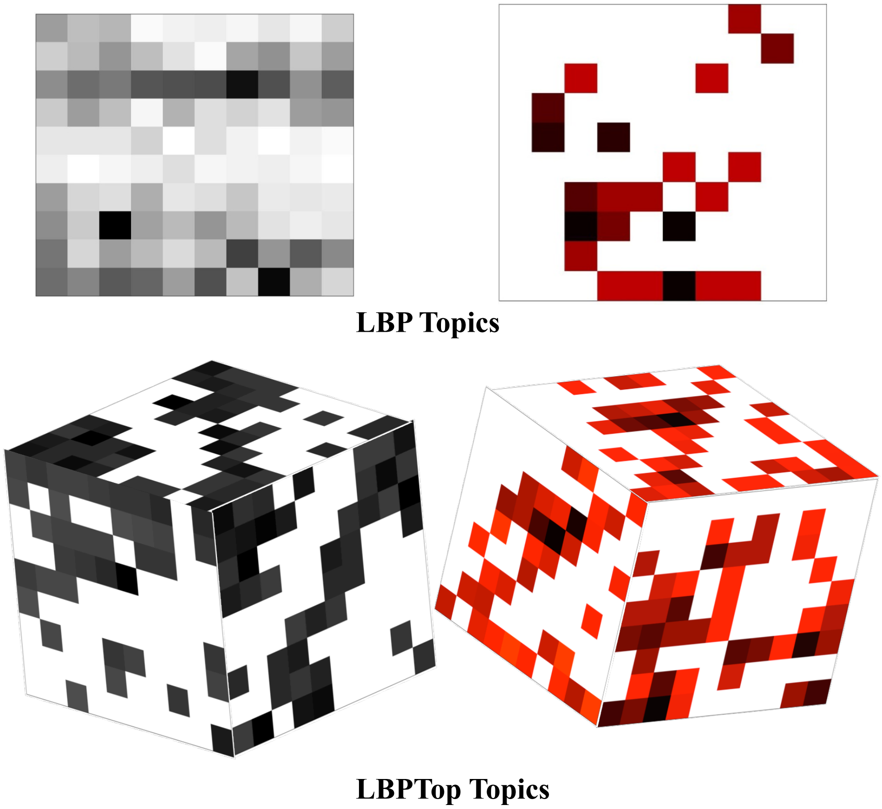

Topic 1
Topic 19
Topic 26
Topic 13
Topic 17
Topic 19
Topic - Term (Feature) Definitions
Document (Image) - Topic Plots

Figure 4.7: Sample plots of Topics extracted using LDA on Landmark features. (*left*) Topic - Term mixture plots i.e. topic definitions and (*right*) Document - Topic mixture plots

top left movement of landmark 35. We have shown some sample plots of landmark based topics extracted from AVEC 2012 dataset in Figure 4.7. In the *left* we have plotted topic definitions where, as described earlier, each topic models the movement of certain points in certain directions. E.g. the first topic models the pulling of left eye brow and similarly the second topic model the opening of mouth. And we realize that these definitions are in fact the Action Units provided by the Facial Action Coding system [23]. In Figure 4.7 (*right*) we have plotted the topics over the face where the point and direction represent a word in the document and their color indicates the topic assigned to them.



LBP Topics

LBPTop Topics

Figure 4.8: 2-D and 3-D illustration of sample topics extracted from LBP and LBPTop features respectively.

Now let us consider the appearance based features LBP and LBPTop which are quantized per block where the block are divided along XY direction for LBP features and XY, XT and YT directions for LBPTop features. Figure 4.8 contains plots of two topics each extracted from LBP and LBPTop features. The color coding of a topic corresponds to the probability of that particular spatio-temporal block e.g. the

Table 4.3: Comparison of Topic features vs Base features on CK+ dataset.

Classifier	Base-SVML	LDA-SVML	LDA-SVMR	LDA-NB	LDA-KNN
Accuracy	66.68%	**85.62%**	84.4%	79.5%	85.32%

darker the block is, the higher the probability that it influences a particular topic. For LBPTop the extension from XY to other dimensions is straightforward and the same interpretation hold to XT and YT dimensions. The plots we have seen until now are topic definitions and some of them may not be intuitive because they are topic-term probabilistic distributions plotted with a spatio-temporal context. While until now we have analyzed topic-term mixtures, in the following section we will consider the second mixture, the document-topic mixtures which are the new features. We will now evaluate these features for their predictive capabilities.

Preliminary Results

Before discussing results from continuous emotion recognition, we briefly experimented with LDA features on Discrete emotion recognition. We have used CKPlus dataset [40] to conduct preliminary analysis and the results that follow are derived using a Gibbs Sampling based LDA model. The CK+ database contains 327 image sequences, annotated with 7 discrete emotions and 34 AUs, from 118 subjects. As in [40], subject wise leave-one-out validation strategy was used in these experiments. Different classifiers like Naive Bayes (NB), K-Nearest Neighbors (KNN), SVML (SVM with Linear kernel) and SVMR (SVM with RBF kernel) have been trained on topic features to predict one of the 7 emotions.

Table 4.3 shows the mean accuracies over all the folds. The base features here are the drifts in the landmark positions and it can be seen that the topics extracted from these base features outperform them irrespective of the classification algorithm used.

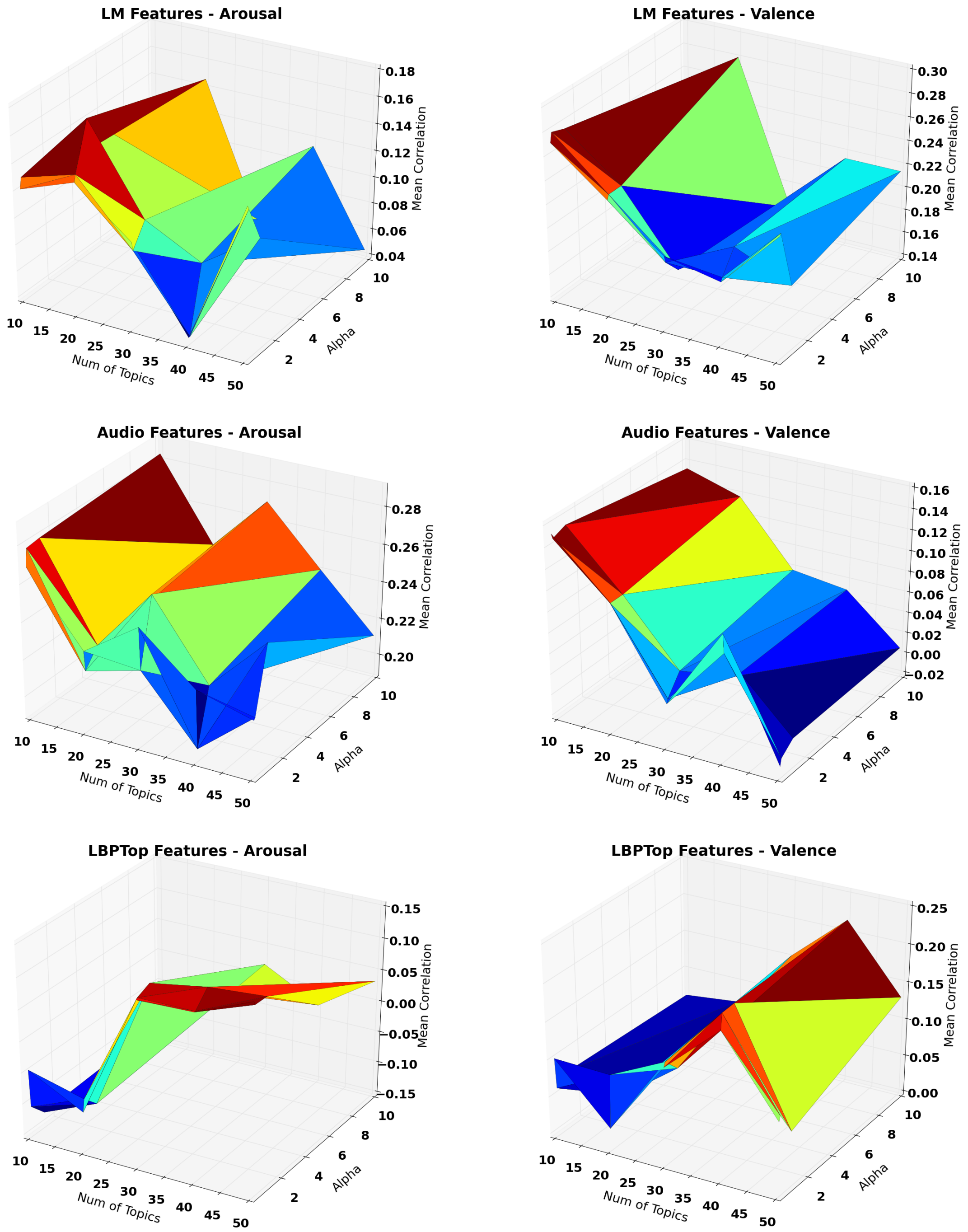


Figure 4.9: Cross Validation results for LDA based topic features on LM, Audio and LBPTop features. Linear regression is used to select the best LDA parameters, $\#$ *of topics* and α's

Table 4.4: Confusion matrix (in %) for 7 emotions using Linear SVM and topic features

	An	Co	Di	Fe	Ha	Sa	Su
An	**82.22**	0.00	8.89	0.00	0.00	8.89	0.00
Co	0.00	**66.67**	11.11	5.56	11.11	5.56	0.00
Di	5.08	0.00	**91.53**	1.69	0.00	1.69	0.00
Fe	4.00	0.00	0.00	**68.00**	8.00	4.00	16.00
Ha	0.00	0.00	0.00	5.80	**92.75**	0.00	1.45
Sa	7.14	7.14	7.14	10.71	0.00	**64.29**	3.57
Su	0.00	1.20	0.00	3.61	0.00	1.20	**93.98**

LDA-SVML i.e. the SVM classifier with linear kernel with topic features performed the best. The confusion matrix for the LDA-SVML model on 7 emotions is shown in Table 4.4 and it can be deduced that *fear* is commonly misclassified as *surprise* in this model, which can be explained by the fact that these two expressions share similar facial movements and hence, similar topics. On the other hand, *contempt* and *sadness* are not expressions of high intensity, and hence are misclassified as other expressions. We will move the discussion to the main set of results where all the LDA models have been evaluated on AVEC12 dataset and have been trained using Variational EM algorithm over all training videos.

Effect of Parameters

Every LDA model is trained using a set of key parameters which are selected using 3-fold cross validation over training data. These parameters are the # of topics K, the Dirichlet priors $\boldsymbol{\alpha}$ and # of EM iterations and they are independently optimized for each base feature. To cover a broad range of parameters we have used # of topics to be {10, 20, 30, 40 ,50} and α's from the set {0.01, 0.1, 1.0, 10.0}. It is to be noted that since we are using a symmetric Dirichlet prior i.e. same α value for all the topics, our α values are scalar. We have used used LDA Variational EM over different combinations of parameters to extract topic features and used Linear regression to predict the continuous emotions *arousal* and *valence* from these features. We have used 3-Fold cross validation over the AVEC12 training videos and selected the parameters that gave the best mean-correlation values (Eq 3.1).

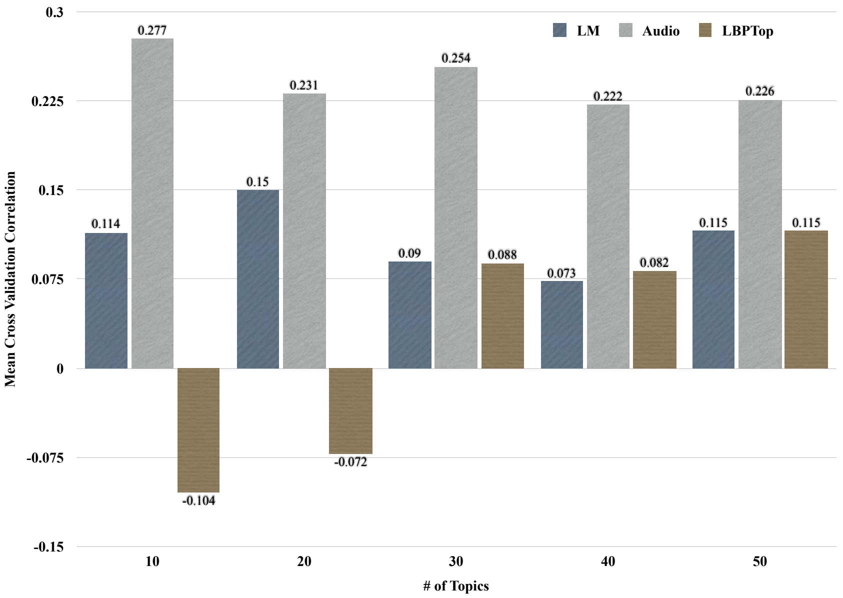


Figure 4.10: Effect of the # of topics on the cross validation performance of LDA topics across features

Figure 4.9 contains detailed plots of the cross validation results from LM, Audio and LBPTop features for both arousal and valence prediction. We observe that the parameters are unique to each feature and the emotion dimension and the # of topics do have an impact on the model performance. Unlike the # of topics, the

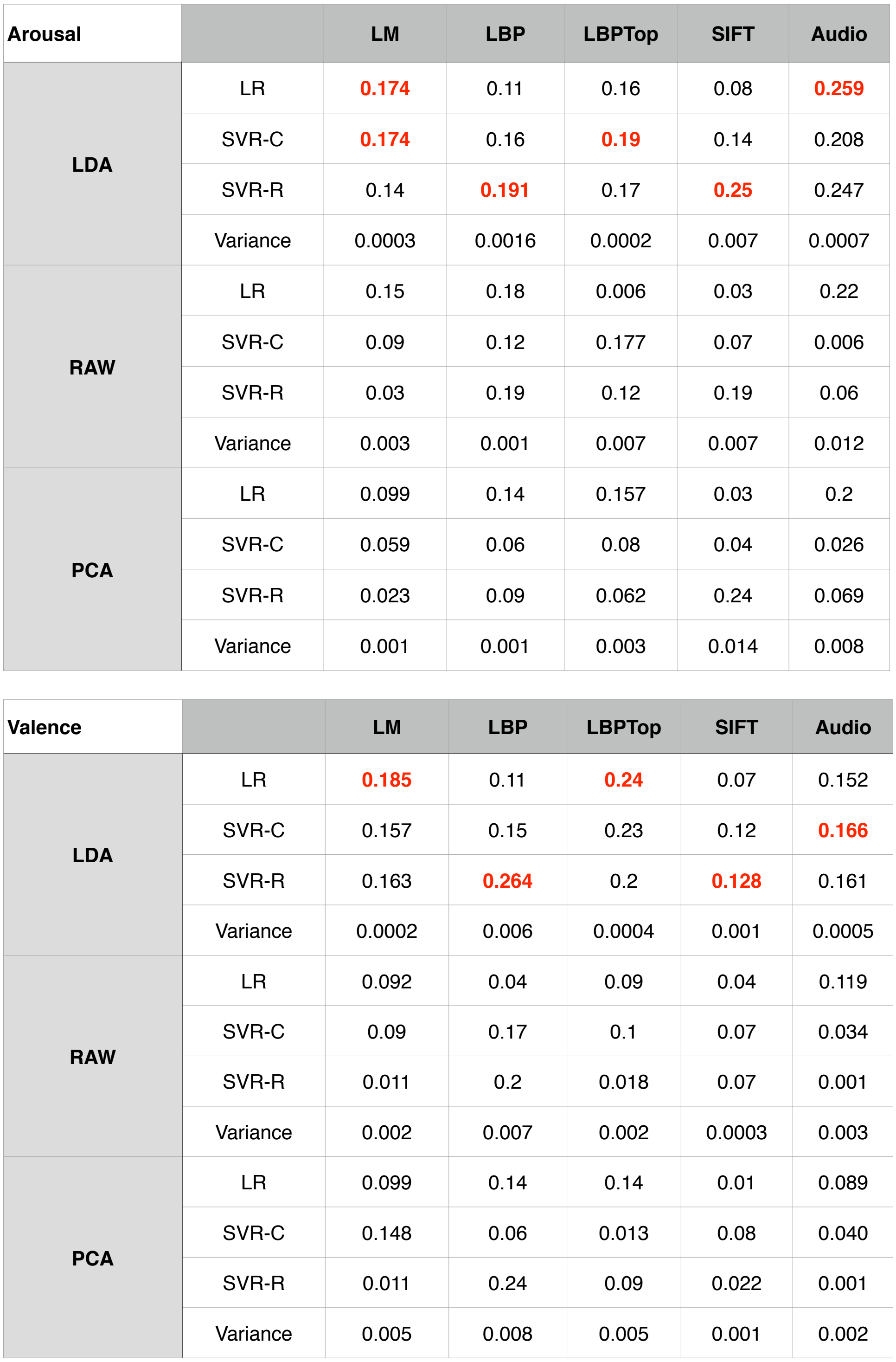
α parameter does not effect the model performance for a given set of topics. We illustrate this in Figure 4.10 where we have plotted only the topic parameters for LM, Audio and LBPTop features and average correlations across different α values. It can bee seen that the # of topics has a predominant affect on the prediction performance and it is very important to chose the best parameters using cross validation. We also calculated mean correlations by varying α's by having topics fixed and the standard deviations for LM, Audio and LBPTop features are 0.029, 0.013 and 0.048 respectively. This implies that the overall change in the performance does not change too much by changing the α parameters. Another important parameter for training an LDA model are the # of EM iterations and throughout this work we have not contained the number of EM iterations but instead continued until the likelihood convergence criteria has been met. Now we will discuss the results of the LDA model on the AVEC12 development videos and evaluate the model in comparison to other techniques.

Comparison of LDA vs PCA

To evaluate LDA features we have used three different regression models viz. linear regression (LR), support vector regression with RBF kernel (SVR-R) and support vector regression with a Cosine kernel (SVR-C). We have trained LDA models using 5 base features LM, LBP, LBPTop, SIFT and Audio features and trained the models using parameters that have been selected with cross validation. We have compared the topic features with the Raw base features and since LDA is a dimensionality reduction technique we have compared it to Principal Component Analysis (PCA) as well. While evaluating PCA we have projected the data onto new space comprising of basis vectors of decreasing priority and whose sum of eigen values account for 98% of the entire sum. The number of features that are reduced after PCA project are 38, 2674, 4865, 4156 and 700 features for LM, LBP, LBPTop, SIFT and Audio features respectively.

Table 4.11 contains the mean cross correlations averaged across all videos for Arousal and Valence prediction. The table contains results across three feature extraction techniques, LDA, Raw (Base) and PCA, where Raw are the actual real values features. For each of the LM, LBP, LBPTop, SIFT and Audio features the correlations from there extraction techniques and three regression models are shown. The row $Variance$ is the variance of three correlations using three regression models, e.g. for the LM feature which has correlations 0.174, 0.174 and 0.14, the variance across three models is 0.0003. Variance is an important factor as it indicates the stability of the features being considered. For arousal prediction, from Table 4.11(*top*) it is evident that LDA based topic features have outperformed the other two base features and also PCA features. The performance of LDA is better than PCA with respect to each of the three regression models individually. Audio based topic features gave the best performance for arousal prediction and among the video features SIFT based topi features performed the best. Among all the features, the most stable topic features are based on LM (0.0003), LBPTop (0.0002) and Audio (0.0007) and the most unstable are the topics from LBP (0.0016) and SIFT (0.007).

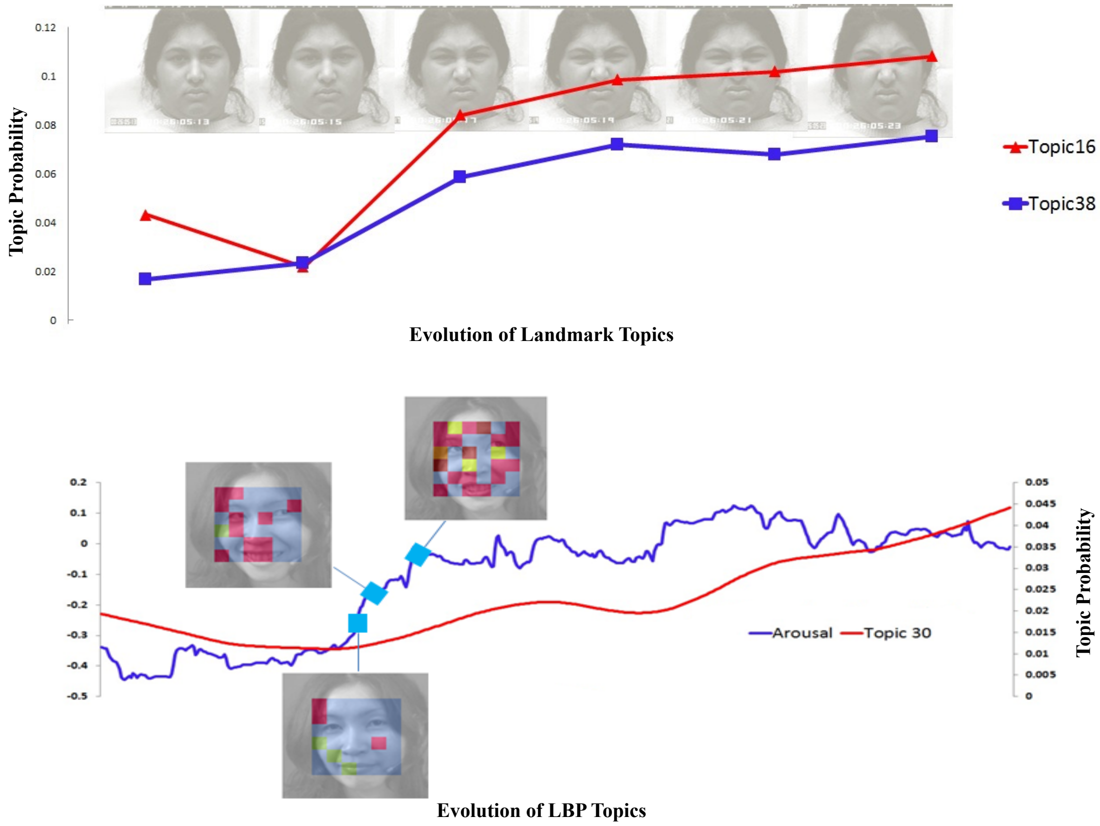
Table 4.11(*bottom*) contains the correlations from Valence prediction and it is again evident that LDA model has again outperformed the Raw and PCA models. The overall variance of LDA model based correlations is 0.002 whereas the variances of correlations from the Raw and PCA models are 0.004 and 0.006 which indicates the stability of the features. As in arousal prediction, for valence, the most stable LDA topics are LM (0.0002), LBPTop (0.0004),and Audio (0.0005) whereas LBP and SIFT features are unstable. The best prediction for valence is given by LBP features and there is a pattern that comes out of these results. The video modality performs



Arousal		LM	LBP	LBPTop	SIFT	Audio
LDA	LR	**0.174**	0.11	0.16	0.08	**0.259**
	SVR-C	**0.174**	0.16	**0.19**	0.14	0.208
	SVR-R	0.14	**0.191**	0.17	**0.25**	0.247
	Variance	0.0003	0.0016	0.0002	0.007	0.0007
RAW	LR	0.15	0.18	0.006	0.03	0.22
	SVR-C	0.09	0.12	0.177	0.07	0.006
	SVR-R	0.03	0.19	0.12	0.19	0.06
	Variance	0.003	0.001	0.007	0.007	0.012
PCA	LR	0.099	0.14	0.157	0.03	0.2
	SVR-C	0.059	0.06	0.08	0.04	0.026
	SVR-R	0.023	0.09	0.062	0.24	0.069
	Variance	0.001	0.001	0.003	0.014	0.008

Valence		LM	LBP	LBPTop	SIFT	Audio
LDA	LR	**0.185**	0.11	**0.24**	0.07	0.152
	SVR-C	0.157	0.15	0.23	0.12	**0.166**
	SVR-R	0.163	**0.264**	0.2	**0.128**	0.161
	Variance	0.0002	0.006	0.0004	0.001	0.0005
RAW	LR	0.092	0.04	0.09	0.04	0.119
	SVR-C	0.09	0.17	0.1	0.07	0.034
	SVR-R	0.011	0.2	0.018	0.07	0.001
	Variance	0.002	0.007	0.002	0.0003	0.003
PCA	LR	0.099	0.14	0.14	0.01	0.089
	SVR-C	0.148	0.06	0.013	0.08	0.040
	SVR-R	0.011	0.24	0.09	0.022	0.001
	Variance	0.005	0.008	0.005	0.001	0.002

Figure 4.11: The Mean Cross correlation across development videos of AVEC12 [52] from LDA models on five different features and three regressors.

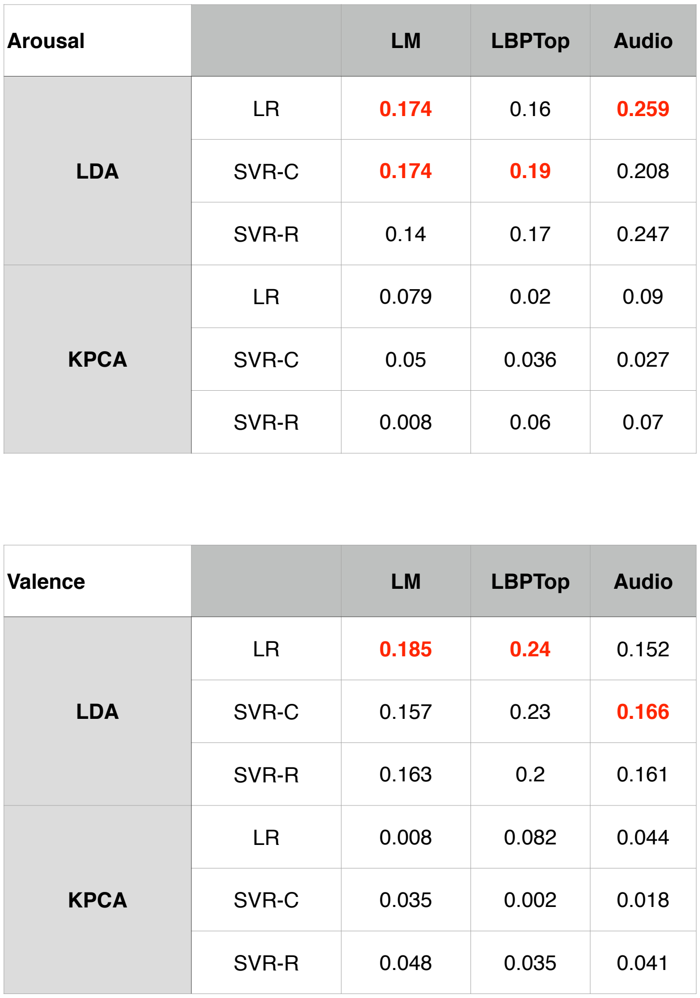


Evolution of Landmark Topics

Evolution of LBP Topics

Figure 4.12: Plots illustrate the evolution of (*top*) landmark topics with discrete emotion *disgust* and (*bottom*) LBP topics with continuous emotion *arousal*

better in predicting valence whereas audio modality performs better when it is arousal prediction. This is an interesting result as this may be indicative of the inherent relationship between modalities and continuous emotions. The reason for why we looked into stability of prediction is to conclusively prove that certain modality and model performs better. It is conclusive from these results that LBP and SIFT features are not very stable when compared to the rest which can be explained by the nature of these features. LBP and SIFT are both non-temporal and spatial features unlike LM, LBPTop and Audio which are in fact temporal features. This also throws some light that for continuous emotion recognition temporal features tend to give more



Arousal		LM	LBPTop	Audio
LDA	LR	**0.174**	0.16	**0.259**
	SVR-C	**0.174**	**0.19**	0.208
	SVR-R	0.14	0.17	0.247
KPCA	LR	0.079	0.02	0.09
	SVR-C	0.05	0.036	0.027
	SVR-R	0.008	0.06	0.07

Valence		LM	LBPTop	Audio
LDA	LR	**0.185**	**0.24**	0.152
	SVR-C	0.157	0.23	**0.166**
	SVR-R	0.163	0.2	0.161
KPCA	LR	0.008	0.082	0.044
	SVR-C	0.035	0.002	0.018
	SVR-R	0.048	0.035	0.041

Figure 4.13: Cross correlation using LDA and KPCA (with RBF Kernel) models on AVEC12 development set videos.

stable performance. For this reason, in rest of the chapters we have thus restricted our evaluation to LM, LBPTop and Audio features only.

Comparison of LDA vs KPCA

Earlier we have compared LDA model performance with that of PCA. LDA can viewed as a non linear projection whereas PCA is a linear projection technique. For this reason we compared the performance of LDA with Kernel PCA (KPCA) which is a non linear projection technique. Kernel PCA was first proposed by Scholkopf et al. [50] which uses the *kernel trick* to transform the data into a higher dimensional space and then project them onto their eigen vector space. Unlike PCA where we calculate the decomposition of covariance matrix, in KPCA we just calculate the final projection using the formula:

$$\Big(\sum_{n=1}^{N} a_n^k \Phi(\boldsymbol{x_n})\Big)^T \Phi(\boldsymbol{x})$$

where k is the k-th principal component onto which the data is projected and N is the total number of data points. We can observe that we actually do not calculate the principal components and are using the Kernel matrix to compute $K(\boldsymbol{x_i}, \boldsymbol{x_j})$. The $\boldsymbol{a}$ vector is calculated using the eigen vector and eigen values of the kernel matrix.

We have built the KPCA model using the Radial Basis Function (RBF) kernel. To compare LDA with KPCA as in PCA we have used top 98% of the eigen space to project the data onto and Table 4.13 contains these results. These results again show that LDA model has outperformed KPCA for all features and regressors which reiterates the power of LDA model.

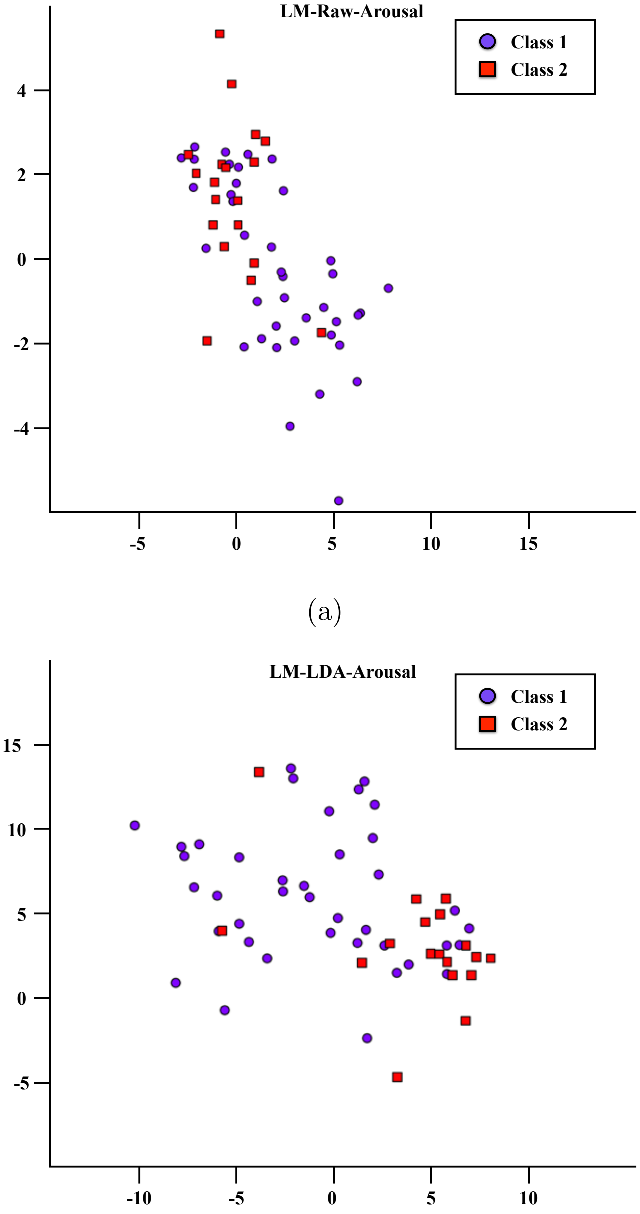
4.1.5 Analysis

In this section we will look into some of the indicators that point out as to why LDA based topics perform well irrespective of the features used. To understand this we plotted the topic features on a human face as an emotion progresses. We studied the evolution of topics with emotions over time and Figure 4.12 illustrates few sample plots. In Figure 4.12(*top*) we have plotted the probability of a single topic over a set of 7 frames of a discrete emotion *disgust*. We have used CohnKanadePlus (CKPlus) [40] database to extract the LM features and trained an LDA model and plotted the probabilities of topics along the intensities of emotions. As shown in the figure, we observed that some of the topics are highly correlated with the intensity of emotions.

Similarly, Figure 4.12(*bottom*) contains the plot of LBP based topic probabilities along with the continuous emotion *arousal*. In the inset we have highlighted the frames and color coded the topic and we again find that there is high correlation between certain topics and emotion. One of the reasons as to why topic features perform better in comparison to traditional dimension reduction technique like PCA is that LDA considers co-occurrence of features and thereby avoids modeling noise.

In general the predictive capability of a dimensionality reduction or feature selection technique depends on whether the features are able to separate the feature space. Figures 4.14 and 4.15 contain plots of two sample videos one for arousal and the other for the valence. Since these emotions are continuous, we considered quantized them into two classes, Class 1 and Class 2 and projected the features to a 2-dimensional space using t-distributed stochastic neighbor embedding (t-SNE) algorithm. Figures 4.14a and 4.14b contain plots of t-SNE projections of the original landmark features and landmark based topic features. We can see that the separability of the feature space with respect to the emotion class is greater in topic features. Similarly Figures 4.15a and 4.15b contain plots for the valence class and it is very clear that the separability of topic features is greater than that of the original feature space.

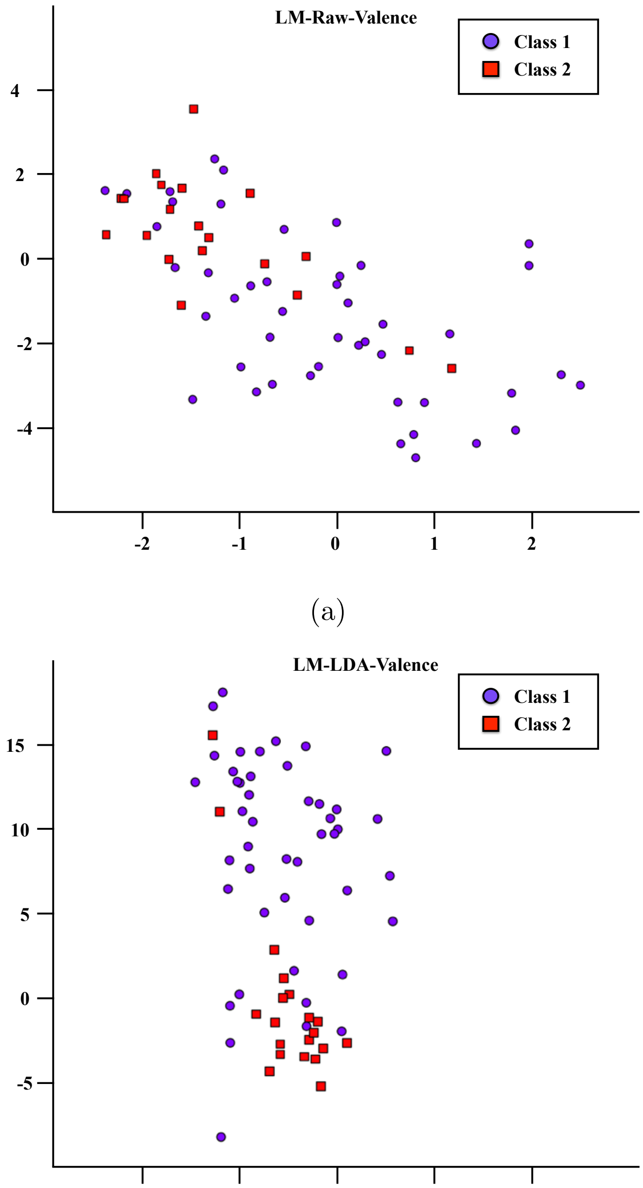
Latent Dirichlet Allocation model can be viewed as a dimension reduction technique where the projection is from a quantized document-term space to a topic feature space. From the results discussed in Section 4.1.4, we can infer that LDA model successfully reduces dimensions into a space that is both semantic as well as predictive. Specific to the context of continuous emotion recognition LDA features have given a better performance with respect to two dimensions *arousal* and *valence*. Unlike PCA



(a)

(b)

Figure 4.14: t-SNE [62] projection of Landmark features (a) and LDA features (b) mapped to two Arousal classes



(a)

(b)

Figure 4.15: t-SNE [62] projection of Landmark features and LDA features mapped to two Valence classes

where the projection is done on the original continuous space, LDA needs a discrete space and thus the features need to be quantized. While trying to retain the inherent capabilities of LDA model, we wanted to find a way to avoid feature quantization. We propose a continuous version of LDA model called the Dirichlet Gaussian Mixture Model (DGMM) where the topic-term mixture is not a Multinomial mixture but a continuous distribution over features. In Section 4.2 we will discuss this model and evaluate its performance against LDA.

4.2 Dirichlet Gaussian Mixture model

LDA model works very well for features that are quantized meaningfully and care has to be taken about the methodology used for quantizing the features. Also, quantizing bag-of-words based video features is quite straightforward but quantizing any other features needs to be well thought. Since almost every multimodal feature is a continuous one, we built and tested a continuous version of the LDA model which we call the Dirichlet Gaussian Mixture Model (DGMM). In this section we show that this continuous model gives a comparable performance as LDA and can be used in scenarios where a proper quantization mechanism cannot be found. Another motivation to build continuous mixture models is that there is always a chance that information is lost during quantization which may decrease the predictive capabilities of the LDA features.

LDA is a double mixture model and we model the continuous double mixture model as an extension to Gaussian Mixture Model (GMM) proposed McLahlan et al. [44]. We will begin our discussion of DGMM by introducing GMM and its inference which come handy in explaining our model. Figure 4.16 contains the graphical models for GMM where $\boldsymbol{f_t}$ is the continuous feature vector of a multimodal document t and in this context $\boldsymbol{\theta_t}$ is the mixture weight vector. And we assume that each of the

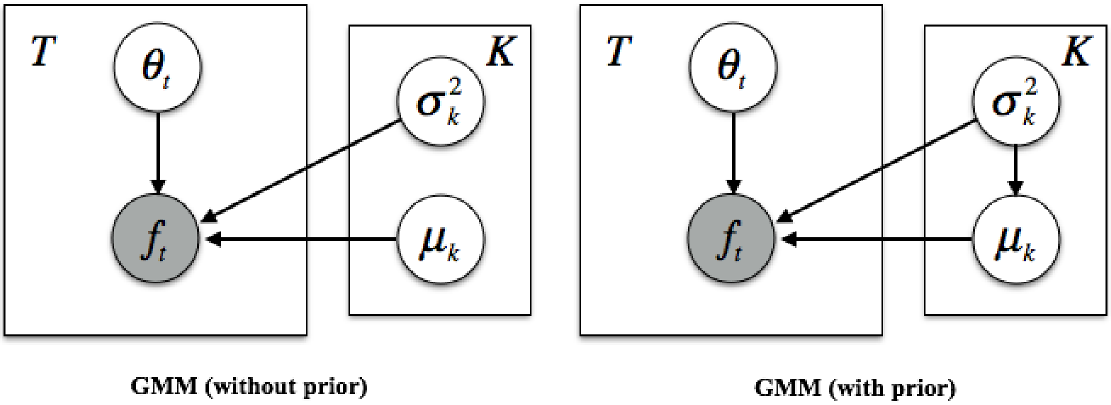


Figure 4.16: Graphical models for Gaussian Mixture Model without prior (*left*) and with prior (*right*)

topics is a Gaussian distribution with a mean μ_k and variance σ_k^2. The significant difference between LDA and GMM is the way a topic is defined, where it represents a Multinomial over words in LDA whereas it represents a Multivariate Gaussian over feature vectors. Another difference is that in LDA each *word feature* is assigned a topic whereas in GMM the entire feature vector is assigned a topic and thus in GMM a multimodal document does not represent a frame but a sequence of frames or a clip. LDA has Dirichlet prior over $\boldsymbol{\theta}$ and as shown in Figure 4.16(*right*), in GMM generally a prior is assumed over the Gaussian parameters. The Gaussian Wishart prior is assumed over these parameters with a conditional dependence introduced between μ_k and variance σ_k^2 and this dependence is indicated as $p(\mu_k, \sigma_k^2) = p(\mu_k)p(\mu_k|\sigma_k^2)$.

As an extension to GMM, Hu et al. [28] proposed Gaussian LDA (GLDA) model which is displayed in Figure 4.17(*left*). In GLDA, instead of modeling a multimodal document, a set of documents are grouped to sequences and each sequence is modeled individually. Each feature vector $\boldsymbol{f_t}$ corresponding to a multi modal document is considered as a *word* and features from all sequences are modeled using K multivariate Gaussian distributions. It can be observed that GLDA is a Bayesian extension to

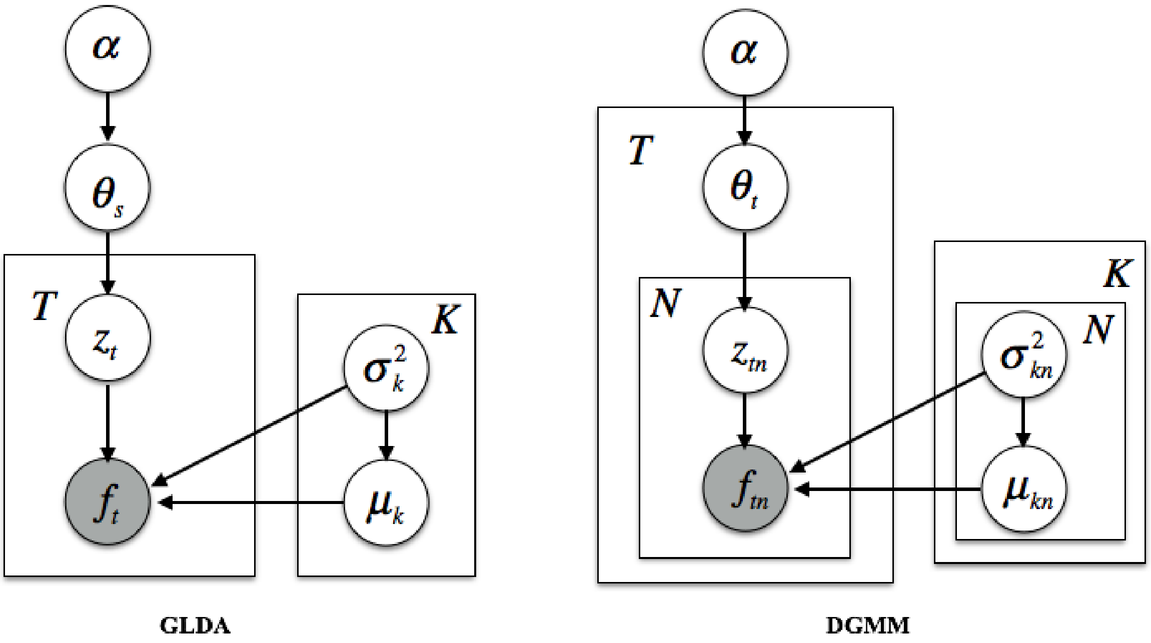


Figure 4.17: Graphical models for Gaussian LDA [28] and Dirichlet Gaussian Mixture Model proposed in this work

GMM with priors on the weight parameters and also the Gaussian parameters. We have proposed a DGMM model where unlike the GLDA model, we propose to model each multimodal document and consider each feature to be a *word*. In DGMM model, there are a total of $K * N$ Gaussian distributions where K is the number of topics and N is the number of features in each continuous multimodal document. Below is the generative process for DGMM:

Generative Process for DGMM:

1. Draw Multinomial K-topic distribution $\boldsymbol{\theta} \sim$ Dirichlet($\boldsymbol{\alpha}$)

2. For each of the N features,

   (a) Assign a topic, $z_n \sim$ Multinomial($\boldsymbol{\theta}$) to n^{th} feature

   (b) Draw a feature $f_n \sim$ Normal($\mu_{z_n,n}, \sigma^2_{z_n,n}$)

We observe that the mean and variance are no longer vectors but are scalars because each feature is associated with a Gaussian distribution. These parameters can in fact be extended to become vectors if we chose to model groups of features. This depends on the features that are being considered, e.g. if the features are audio LLD features, then since each one is self descriptive, modeling each feature individually is meaningful. But if the feature vector is a concatenation of histograms, then it makes sense to group all the features within the histogram.

4.2.1 Variational EM Algorithm for DGMM

The likelihood of observing a document feature vector $\boldsymbol{f_t}$ given the Dirichlet and Gaussian parameters is:

$$p(\boldsymbol{f_t}|\boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\sigma^2}) = \int_{\boldsymbol{\theta_t}} p(\boldsymbol{\theta_t}|\alpha) \prod_{n=1}^{N} \sum_{k=1}^{K} p(f_{tn}|\mu_{kn}, \sigma^2_{kn}) p(k|\boldsymbol{\theta}_t) \boldsymbol{d\theta_t} \tag{4.13}$$

Similarly, the posterior distribution is given by:

$$p(\boldsymbol{\theta}, \boldsymbol{z}|\boldsymbol{f}, \boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\sigma^2}) = \frac{p(\boldsymbol{\theta}, \boldsymbol{z}, \boldsymbol{f}, \boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\sigma^2})}{p(\boldsymbol{f}|\boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\sigma^2})}$$

As in LDA, the likelihood and the posterior distributions are intractable and so we will use Variational EM algorithm to deterministically infer the DGMM parameters. Figure 4.4 very well defines the simplified version of DGMM model as the dependence between the features and Gaussian parameters is dropped. We assume a surrogate distribution q that corresponds to the simplified model in Figure 4.4 to approximate the above posterior. We aim to minimize the KLDivergence between the surrogate and original posterior distribution which is equivalent to maximizing the Expected Lower Bound, $\boldsymbol{\mathcal{L}}(\boldsymbol{\gamma}, \boldsymbol{\phi}; \boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\sigma^2})$, which is given by Eq 4.8. The solution for DGMM differs from the one in Eq 4.8 only through the term $E_q[\log(p(\boldsymbol{f}|\boldsymbol{z}, \boldsymbol{\mu}, \boldsymbol{\sigma^2}))]$ instead of $E_q[\log(p(\boldsymbol{v}|\boldsymbol{z}, \boldsymbol{\pi}))]$. The expected lower bound for DGMM is given by:

Algorithm 5 Variational Bayes algorithm for DGMM

```
Input: All multimodal document features $\boldsymbol{f}$ and Dirichlet Priors $\boldsymbol{\alpha}$ , # of Topics $K$, # of EM iterations $I$, # of E iterations $J$
Parameters: Gaussian parameters $\boldsymbol{\mu}_0$ and $\boldsymbol{\sigma^2}_0$
Output: Document-Topic distributions $\boldsymbol{\gamma}$ and Topic-Feature distributions $\boldsymbol{\mu}$ and $\boldsymbol{\sigma^2}$
Initialize the parameters using K Means with $k$ clusters on each feature $n$.
$i := 0$
procedure DGMMVBAlgorithm
    while $i < I$ or $!converged$ do                ▷ Iterate till convergence
        E-Step:
        for each document $t$ do
            Initialize equal values to all topics in $\boldsymbol{\gamma_t}$ as $\gamma_{tk} = \alpha_k + \frac{N_t}{K}$
            $j := 0$
            while $j < J$ or $!converged$ do
                (i) Calculate the Multinomials $\boldsymbol{\phi_t}$ using Eq 4.16
                (ii) Calculate the Dirichlet priors $\boldsymbol{\gamma_t}$ using Eq 4.11
                $j := j + 1$
            end while
        end for
        M-Step:
        (i) Estimate Gaussian means $\boldsymbol{\mu}_i$ using Eq 4.18
        (i) Estimate Gaussian variances $\boldsymbol{\sigma^2}_i$ using Eq 4.19
        $i := i + 1$
    end while
end procedure
```

$$\begin{aligned}\mathcal{L}(\boldsymbol{\gamma},\boldsymbol{\phi};\boldsymbol{\alpha},\boldsymbol{\pi}) &= E_q[\log(p(\boldsymbol{\theta}|\boldsymbol{\alpha}))] + E_q[\log(p(\boldsymbol{z}|\boldsymbol{\theta}))] + E_q[\log(p(\boldsymbol{f}|\boldsymbol{z},\boldsymbol{\mu},\boldsymbol{\sigma^2}))] \\ &\quad -E_q[q(\boldsymbol{\theta})] - E_q[q(\boldsymbol{z})] \qquad (4.14)\end{aligned}$$

where the term $E_q[\log(p(\boldsymbol{f}|\boldsymbol{z},\boldsymbol{\mu},\boldsymbol{\sigma^2}))]$ is given by

$$E_q[\log(p(\boldsymbol{f}|\boldsymbol{z},\boldsymbol{\mu},\boldsymbol{\sigma^2}))] = \sum_n \sum_k q(z_{nk}) \log(\mathcal{N}(f_n|\mu_{kn},\sigma^2_{kn}))$$

Maximizing the function $\mathcal{L}$ with respect to $\boldsymbol{\phi}$ with the constraint that $\sum\limits_k \phi_{nk} = 1$ we arrive at:

$$\begin{aligned}\phi^*_{tnk} = \underbrace{\arg\max}_{\phi_{tnk}} \;\; &\{\phi_{tnk}(\Psi(\gamma_{nk}) - \Psi(\textstyle\sum \gamma_{nk})) + \phi_{tnk}\log(\mathcal{N}(f_t n|\mu_{kn},\sigma^2_{kn})) \\ &-\phi_{tnk}\log(\phi_{tnk})\} + \lambda_{tn}(\sum_k \phi_{tnk} - 1) \qquad (4.15)\end{aligned}$$

where t, n, k represent the document, features and topic respectively. Differentiating the above equation with ϕ_{tnk} and equating it to zero we obtain the optimal ϕ_{tnk} as:

$$\phi^*_{tnk} \propto \mathcal{N}(f_{tn}|\mu_{kn},\sigma^2_{kn}) \exp(\Psi(\gamma_k) - \Psi(\sum \gamma_k)) \qquad (4.16)$$

In order to obtain the optimal values for $\boldsymbol{\mu}$ we maximize Eq 4.14 with respect to $\boldsymbol{\mu}$ as given below:

$$\begin{aligned}[\mu^*_{kn},\sigma^{2^*}_{kn}] &= \underbrace{\arg\max}_{\mu_{kn},\sigma^2_{kn}} \sum_t \sum_n \sum_k \{\phi_{tnk}\log(\mathcal{N}(f_t n|\mu_{kn},\sigma^2_{kn}))\} \\ &\sim \underbrace{\arg\max}_{\mu_{kn},\sigma^2_{kn}} \sum_t \sum_n \sum_k \Big\{\phi_{tnk}\Big(\log(\sigma^2_{nk}) + \frac{(f_{tn}-\mu_{kn})^2}{2\sigma^2_{kn}}\Big)\Big\} \qquad (4.17)\end{aligned}$$

We observe that the summation constraint over the topic probabilities $p(f|\mu,\sigma^2)$ is no longer enforced because we are dealing with a continuous Gaussian distribution whose cumulative density function ensures that $\int\limits_{-\infty}^{\infty} \mathcal{N}(f|\mu_{kn},\sigma^2_{kn})df = 1$ for all k.

Calculating the partial derivative of the function in Eq 4.17 with respect to μ_{kn} and equating it to zero we have:

$$\sum_t \phi_{tnk} \frac{(f_{tn} - \mu_{kn})}{\sigma^2_{kn}} = 0$$

$$\implies \mu^*_{kn} = \frac{\sum\limits_t \phi_{tnk} f_{tn}}{\sum\limits_t \phi_{tnk}} \tag{4.18}$$

The above formula has an intuitive explanation that the mean of a feature within a topic is a weighted sum of the feature values weighted by their topic assignment probabilities. Calculating the partial derivative of the function in Eq 4.17 with respect to σ^2_{kn} and equating it to zero we have:

$$\sum_t \phi_{tnk} (\frac{1}{\sigma^2_{kn}} - \frac{(f_{tn} - \mu_{kn})^2}{2(\sigma^2_{kn})^2}) = 0$$

$$\implies \sigma^{2*}_{kn} \propto \frac{\sum\limits_t \phi_{tnk} (f_{tn} - \mu_{kn})^2}{\sum\limits_t \phi_{tnk}} \tag{4.19}$$

Again the intuitive explanation to this formula is that the variance is a weighted sum of the individual variances weighted by the topic assignment probabilities.

It is important to note that in the the derivation of values for μ and σ^2 in GLDA [28], the authors made a mistake in deriving this formula where they ignored the normalizing factor. This factor is extremely important as the results change drastically if the formulae are not normalized. We observe that the calculating of ϕ_{tnk} using Eq 4.16 is the E-step in the variational inference. Since we need the ϕtnk values to calculate the Gaussian parameters, calculating μ_{nk} and σ^2_{nk} becomes the M-step of the variational EM algorithm. While calculating the σ^2_{nk} value, the μ_{nk} value calculated in the current iteration is used and not the one from the previous iteration.

Algorithm 5 puts together the entire Variational EM algorithm for DGMM model and as in LDA model we iterate until convergence or for a fixed number of iterations.

The quality of the optimal parameters and topic assignments given by DGMM depend a lot on the initial parameters. Algorithm 6 shows the steps in the initialization procedure of DGMM. We consider all feature values of n^{th} feature $\boldsymbol{f_n}$ and perform K-Means [42] clustering using K clusters where K is the number of topics used in the DGMM model. After clustering, we extract all the n^{th} feature values that have been assigned to k^{th} cluster as $\boldsymbol{f_{nk}}$. A Gaussian distribution is fit to these feature values and consequently μ_{nk} and σ_{nk}^2 become the Gaussian estimates. This initialization is very efficient in comparison to the initialization in LDA where topic assignment is done using a Uniform distribution.

Algorithm 6 Algorithm to initialize parameters of DGMM model

```
Input: All multimodal document features $\boldsymbol{f}$, # of Topics $K$
Output: Gaussian parameters $\boldsymbol{\mu}$ and $\boldsymbol{\sigma^2}$
procedure InitializeDGMM
    for each feature $n$ do
        Cluster the feature values $\boldsymbol{f_n}$ to $K$ cluster using K-Means clustering
        Obtain $\boldsymbol{f_{nk}}$ as feature values assigned to topic $k$
        for each topic $k$ do
            Estimate $\mu_{nk}$ and $\sigma_{nk}^2$ by fitting a Gaussian distribution on $\boldsymbol{f_{nk}}$
        end for
    end for
end procedure
```

In equations Eq 4.10 and Eq 4.16 we observe that estimation of the ϕ values needs the calculation of $p(v_{tn}|\pi_k)$ and $p((f_{tn}|\mu_{kn}, \sigma_{kn}^2)$. It is straightforward to calculate these for LDA because v_tn belong to a finite vocabulary and thus estimating $p(v_{tn}|\pi_k)$ is dependent only of the size of the vocabulary V and need not be calculated for each document. But in the case of DGMM this probability of each feature has to be calculated for each document as we are dealing with continuous features. And thus these calculations are dependent on the total number of documents and Appendix B.3 shows a snapshot of this calculation during the M-step so that they can be reused in the E-step. There is a possibility that the log probability could be infinity in cases where the feature may be an outlier or has not been observed before and in such cases we assign an extremely low probability to that feature.

The total space complexity of the algorithm is of the order $O((T * N) + (T * K * N) + (K * N))$ where T, N and K are the number of documents, features and topics respectively. Since K is far smaller than T and N the complexity approximates to $O(T * K * N)$ which is greater than that of LDA which is $O(K * V)$. This is due to the fact that the Gaussian probabilities are precalculated and stored as shown in Appendix B.3. The time complexity of DGMM algorithm is of the order $O(I * ((T * J * N * K) + (T * N * K)))$ which is also greater than that of LDA as we are dealing with continuous features instead of a fixed vocabulary. As in LDA we have implemented DGMM algorithm using Map Reduce framework there by parallelizing the E-step over the training documents. As discussed earlier we address the parameter estimation as training because we use documents from a training set to learn DGMM model. During testing, we infer the document-topic distributions on unseen or test documents by using the same EM algorithm 5 but by only running the E step until convergence.

4.2.2 Experiments and Results

Comparison of DGMM vs LDA:

We have evaluated DGMM model on two criteria, 1) its predictive performance in comparison to LDA model, and 2) its algorithmic complexity in comparison to LDA model. We considered these criteria because there is a trade off between the complexity and performance of a model. In Figure 4.18 we have plots of the log likelihood changes for LDA and DGMM models and it is interesting to observe that the convergence of DGMM EM algorithm is much faster than that of LDA e.g. in this figure, LDA takes 10 iterations to converge whereas DGMM coverages almost after 5 iterations. Another interesting observation is that the rate of change of likelihood in DGMM is pretty small right from the second iteration. The probable explanation to this is that since we are using Gaussian distributions, once the the optimal parameters are being set in an iteration, not much change is occurring to them over the next few iterations. This can be a case of the model clinging to local optimal values which is a matter of concern to us.

Table 4.19 contains the results from DGMM in comparison to LDA and PCA models. We have used AVEC12 development videos for our evaluation and we three different features, viz.LM, LBPTop and Audio. We have chosen these three features as discussed in Section 4.1.4, these features have given the most stable results. The results shown in Table 4.19 are the cross correlations from a Linear regression model. The take away from these results is that DGMM has given a comparable if not better results when compared to LDA model. DGMM performed well for the LM and LBPTop features for valence and arousal respectively and gave comparable performance to LDA for video modality. In the case of the audio modality, DGMM has not performed as well as LDA (not very comparable) which indicates the value of

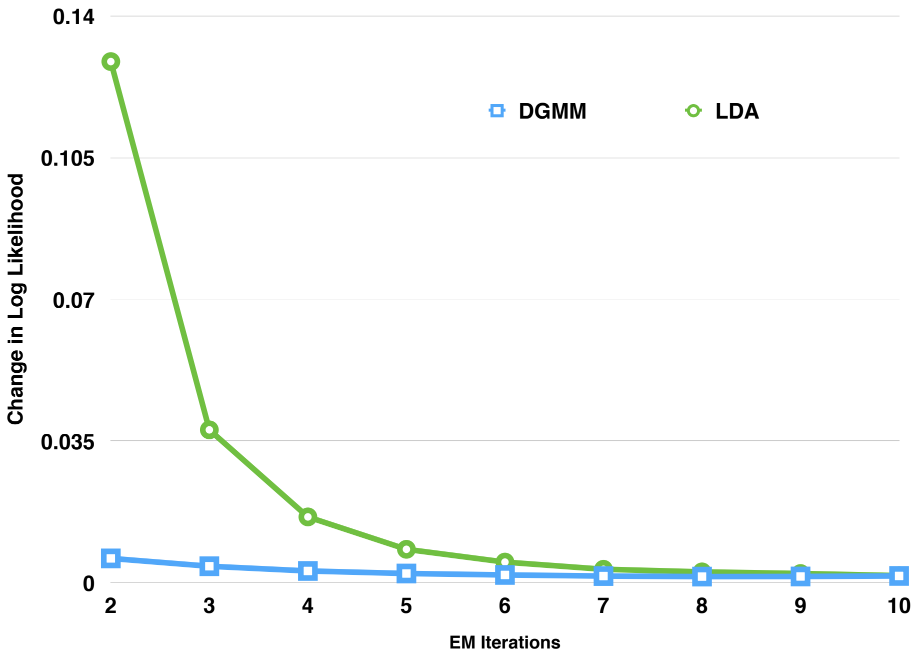
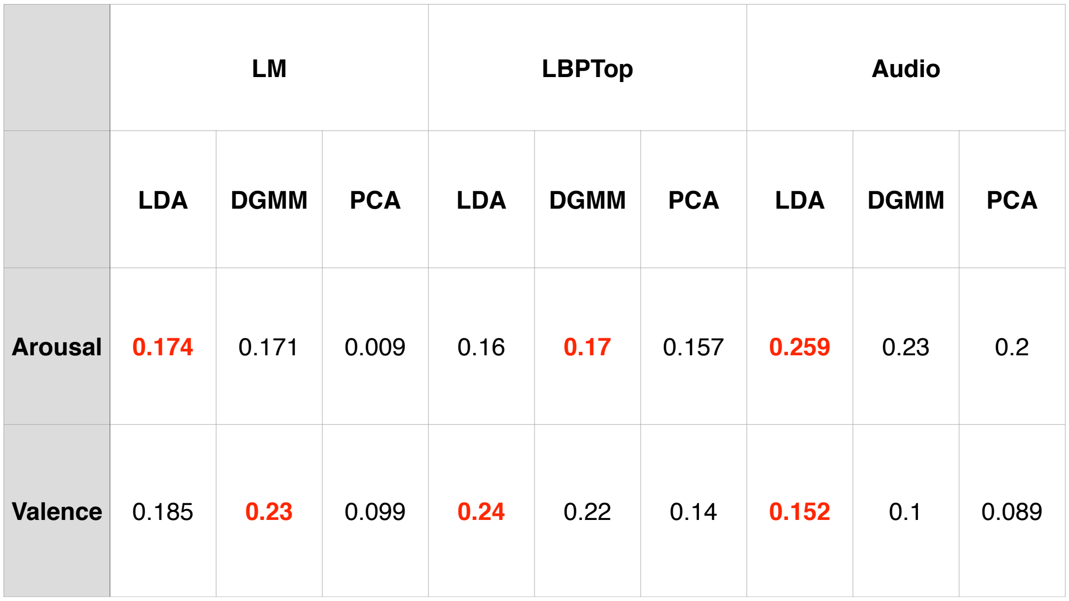


Figure 4.18: Comparison of changes in log likelihood values after each EM iteration for LDA and DGMM

quantization especially for audio features. Immaterial of the modality, DGMM has outperformed PCA model which is also a dimension reduction technique that operates on continuous feature space.

We have evaluated the complexity of DGMM as a function of the time taken to train the model. In Figure 4.20, we have plotted the times taken by EM algorithm to train DGMM and LDA models across different features and topics. These run times are in minutes and we have gathered them from an Intel i7 3.4GHz, 12 core processor where the EM algorithm has been parallelized to run on 12 cores. We observe that DGMM takes a bit longer to complete training when compared to LDA and it increases with the size of topics. The reason the models take highest time for LBP top is due to its large dimension size. We have to note that the execution time



	LM			LBPTop			Audio		
	LDA	DGMM	PCA	LDA	DGMM	PCA	LDA	DGMM	PCA
Arousal	**0.174**	0.171	0.009	0.16	**0.17**	0.157	**0.259**	0.23	0.2
Valence	0.185	**0.23**	0.099	**0.24**	0.22	0.14	**0.152**	0.1	0.089

Figure 4.19: Comparison of performances of LDA and DGMM models using mean cross correlations across AVEC12 dataset. Linear Regression has been used for this evaluation.

for PCA is much slower than DGMM and in that point of view also DGMM is a better reduction technique when compared to PCA models.

From these results it is conclusive that DGMM is a better dimension reduction technique than PCA both interns of performance and execution time. When compared to LDA, DGMM has given a comparable performance but does not scale as well as LDA does with increasing feature size and topic size. The reason behind the longer execution times in DGMM are due to the calculation of the probability densities for each feature in each document after every EM step. Unlike in LDA the probability is

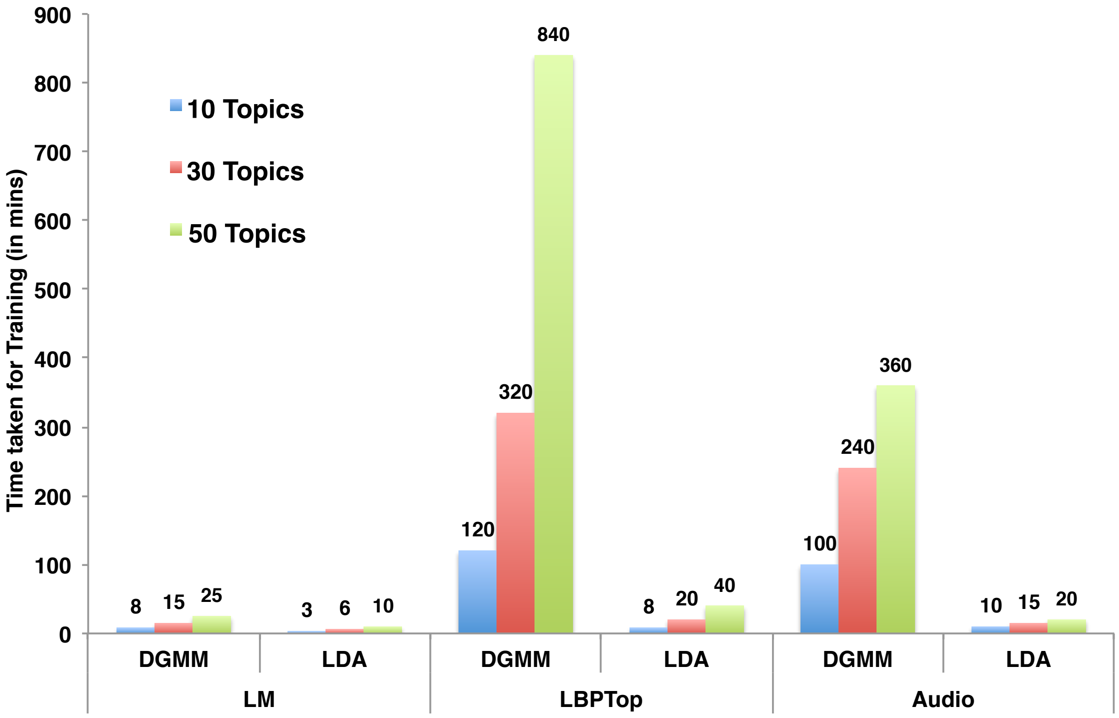


Figure 4.20: Comparison of time taken to train LDA and DGMM models for different features and topics.

a normalized frequency matrix, in DGMM the probability density is continuous and has to be evaluated for every feature under consideration.

4.3 Conclusions

In this chapter we have introduced the Expectation Maximization framework using the graphical structure of unsupervised topic models specifically Latent Dirichlet Allocation. We have derived and discussed the Variational inference techniques for two unsupervised models, LDA and DGMM. We have proved our hypothesis that LDA model based feature extraction generates meaningful visualizable topics whose performance is better than traditional dimension reduction techniques. We have evaluated models across features and regressors and analyzed the reasons behind the good performance of LDA models. Since LDA requires feature quantization and there may be loss of information in that process, we proposed a double mixture model based on the framework of LDA called the Dirichlet Gaussian Mixture Model (DGMM). We have derived the inference of different parameters of DGMM using Variational technique. DGMM is a mixture model with Multinomial and Gaussian mixture distributions and is assumes a Dirichlet prior over the Multinomial. We have specifically evaluated these unsupervised models on Continuous emotion recognition which is generalizable to any multimodal time series application.

From our evaluation we glean that since the same topic features are used for both *arousal* and *valence*, they tend to model either dimension well but not both. This is due to the fact that the topics may not necessarily contain feature correlations that are valuable for emotion analysis but may also contain other information like the texture or facial indicators of the person himself. In order to specifically learn topics that are aligned towards emotion, in chapter 5 we discuss supervised extensions to topic extraction. We will introduce supervised LDA model and its inference and propose new models that do not overfit and do consider multi modal and multi feature interactions.

Chapter 5

SUPERVISED TOPIC MODELS

In Chapter 4 we discussed how unsupervised probabilistic topic models can be trained and used in emotion recognition. These models can be perceived as feature extraction or dimension reduction techniques. Different classification and regression models have been used with topic features to perform the predictions of emotions. In unsupervised topic models, the feature extraction phase is not effected by the emotion values associated with documents. In this chapter we discuss existing and new supervised topic models where the feature extraction or dimension reduction is effected by the emotion labels. These are called the supervised topic models as the emotion label is also part of their graphical structure and the quality of topics extracted is influenced by the emotions that are being modeled.

5.1 Supervised Latent Dirichlet Allocation model

Topic models are based on Latent Semantic Indexing models which can be looked at as dimension reduction or projection techniques equivalent to PCA but for discrete data. There has been some research on supervised versions of PCA e.g. Bair et al. [9] propose a supervised version by performing PCA on only a subset of features that are most informative. Barshan et al. [10] have proposed another supervised method but it is restricted to classification where the class labels are categorical. Not much attention has been given to the discrete counterpart of PCA and also in the context of a regression model. The primary reason is that LSI models are more often used for information and document retrieval than to predict something unlike PCA which is a very popular technique in prediction modeling. But supervised versions of the probabilistic topic models have come in soon after the unsupervised LDA models have been proposed. Interestingly the context in which the first supervised topic model has been proposed is in computer vision [17] where SLDA has been used to automatically annotate images. But later on, different supervised models have been proposed in the context of text analysis and ratings prediction [12]. Even though the earlier versions of SLDA model only dealt with regression models, [17] have proposed a supervised topic model for classification where labels are discrete in nature. The major distinction between supervised models for regression and classification lies in the assumption of the distribution used to model the response variables. E.g Gaussian and Poisson distributions are used to model continuous variables whereas a softmax [6] function is used in discrete cases.

Since we are mostly interested in modeling continuous responses, the discussion in this chapter is based on the work published by Blei [12]. Figure 5.1(*top*) is a representation of the graphical model of SLDA and each of the random variables are explained in Figure 5.1(*bottom*). The entire graphical structure is similar to LDA model except for the new random variable $\boldsymbol{y_t}$ and a new parameter $\boldsymbol{b}$ which represent the continuous response labels and regression coefficients respectively. From Figure 5.1 we observe that the response variable y_t of a given document t is dependent on the topic assignments $\boldsymbol{z_t}$. This implies that the normalized empirical distribution $\boldsymbol{\tilde{z}_t}$ given below is responsible for the response.

$$\boldsymbol{\tilde{z}_t} = \frac{1}{N_t} \sum_n \boldsymbol{I_{nw}}$$

$$\text{where } I_{nw} = \begin{cases} 1 \text{ if } n = w \\ \\ 0 \text{ if } n \neq w \end{cases}$$

It is interesting to note that the response variable y_t is not dependent on the generic Multinomial distribution $\boldsymbol{\theta_t}$ but instead on the empirical topic distribution $\boldsymbol{\tilde{z}_t}$. This is

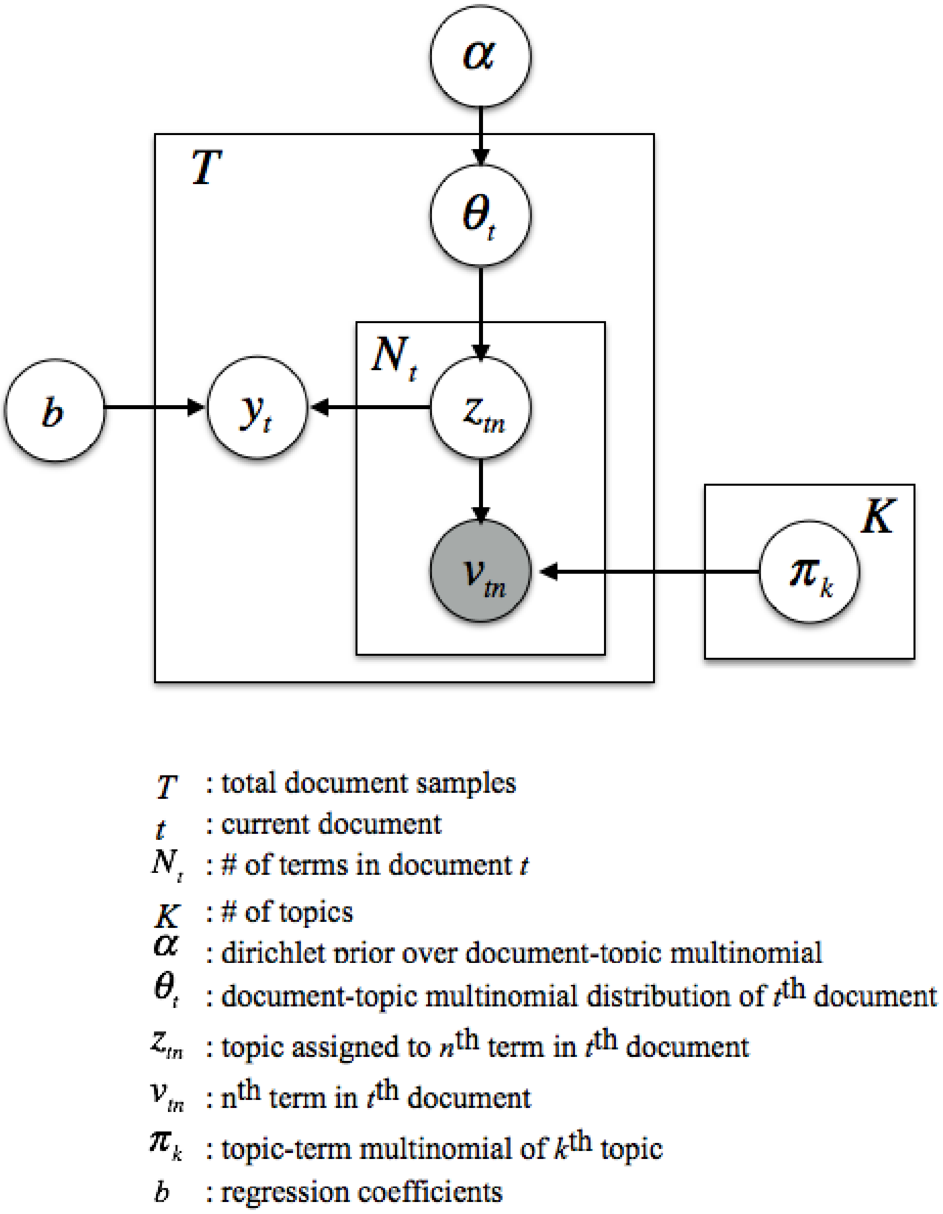

α
T
θ_t
N_t
b
y_t
z_{tn}
v_{tn}
K
π_k
T : total document samples
t : current document
N_t : # of terms in document t
K : # of topics
α : dirichlet prior over document-topic multinomial
θ_t : document-topic multinomial distribution of t^{th} document
z_{tn} : topic assigned to n^{th} term in t^{th} document
v_{tn} : n^{th} term in t^{th} document
π_k : topic-term multinomial of k^{th} topic
b : regression coefficients

Figure 5.1: Graphical model for Supervised Latent Dirichlet Allocation (SLDA) model

to emphasize that the response variable is realistically more aligned to the distribution specific to the document rather than a generalized distribution. The dependency between $\tilde{\boldsymbol{z}}_{\boldsymbol{t}}$ and y_t is modeled using a Generalized Linear Model (GLM) with a linear combination $\boldsymbol{b}^T\tilde{\boldsymbol{z}}_{\boldsymbol{t}}$ where $\boldsymbol{b}$ are the predictor coefficients. A GLM is an exponential probability distribution family defined specifically as a function of a linear combination $\boldsymbol{b}^T \tilde{\boldsymbol{z}}_{\boldsymbol{t}}$ and a dispersion parameter δ and is given by:

$$p(y_t|\tilde{\boldsymbol{z}}_{\boldsymbol{t}}, \boldsymbol{b}, \delta) = h(y_t, \delta) \exp\left\{ \frac{(\boldsymbol{b}^T \tilde{\boldsymbol{z}}_{\boldsymbol{t}}) y_t - A(\boldsymbol{b}^T \tilde{\boldsymbol{z}}_{\boldsymbol{t}})}{\delta} \right\}$$

where h and A are functions of y_t, δ and $\tilde{\boldsymbol{z}}_{\boldsymbol{t}}$ depending on the distribution that is assumed. The most popular distributions that fit into GLM family are the Gaussian and Poisson distributions where GLM becomes linear regression for the former and poisson regression for the latter. In this context since SLDA is also a generative model, we will look into the generative process:

Generative Process for SLDA:

1. Draw a Multinomial K-topic distribution $\boldsymbol{\theta} \sim \text{Dirichlet}(\boldsymbol{\alpha})$

2. For each of the N words,

   (a) Assign a topic, $z_n \sim \text{Multinomial}(\boldsymbol{\theta})$ to n^{th} word

   (b) Draw a term $w_n \sim \text{Multinomial}(\boldsymbol{\pi}_{z_n})$

3. Draw the response variable $y_t \sim \text{GLM}(\tilde{\boldsymbol{z}}_{\boldsymbol{t}}, \boldsymbol{b}, \delta)$

The joint distribution of SLDA model is given by:

$$\begin{aligned} p(\boldsymbol{\theta_t}, \boldsymbol{z_t}, \boldsymbol{v_t}, y_t, \boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{b}) &= p(\boldsymbol{\theta_t}|\boldsymbol{\alpha}) p(\boldsymbol{z_t}|\boldsymbol{\theta_t}) p(\boldsymbol{v_t}|\boldsymbol{z_t}, \boldsymbol{\pi}) p(y_t|\boldsymbol{z_t}, \boldsymbol{b}) \\ &= p(\boldsymbol{\theta_t}|\alpha) \Big(\prod_{n=1}^{N_t} p(v_{tn}|\boldsymbol{\pi}_{z_{tn}}) p(z_{tn}|\boldsymbol{\theta}_t) \Big) p(y_t|\boldsymbol{z_t}, \boldsymbol{b}) \qquad (5.1) \end{aligned}$$

The likelihood of generating the observed variables $\boldsymbol{v_t}$ and y_t for a document t is given by integrating or summing over the hidden variables as:

$$p(\boldsymbol{v_t}, y_t|\boldsymbol{\alpha}, \boldsymbol{\pi}, \boldsymbol{b}) = \int_{\boldsymbol{\theta_t}} p(\boldsymbol{\theta_t}|\alpha) \Big(\sum_{\boldsymbol{z_t}} \Big(\prod_{n=1}^{N_t} p(v_{tn}|\boldsymbol{\pi}_{z_{tn}}) p(z_{tn}|\boldsymbol{\theta_t}) \Big) p(y_t|\boldsymbol{z_t}, \boldsymbol{b}) \Big) \boldsymbol{d\theta_t} \qquad (5.2)$$

Using the above formula the posterior of the latent variable given the observed variable and parameters is given by:

$$\begin{aligned}
p(\boldsymbol{\theta_t}, \boldsymbol{z_t}|\boldsymbol{v_t}, y_t, \boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{b}) &= \frac{p(\boldsymbol{\theta_t}, \boldsymbol{z_t}, \boldsymbol{v_t}, y_t, \boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{b})}{p(\boldsymbol{v_t}, y_t|\boldsymbol{\alpha}, \boldsymbol{\pi}, \boldsymbol{b})} \\
&\quad \text{substituting Eq 5.1 and Eq 5.2 we have} \\
&= \frac{p(\boldsymbol{\theta_t}|\alpha)\Big(\prod_{n=1}^{N_t} p(v_{tn}|\boldsymbol{\pi}_{z_{tn}})p(z_{tn}|\boldsymbol{\theta}_t)\Big)p(y_t|\boldsymbol{z_t}, \boldsymbol{b})}{\int_{\boldsymbol{\theta_t}} p(\boldsymbol{\theta_t}|\alpha)\Big(\sum_{\boldsymbol{z_t}}\Big(\prod_{n=1}^{N_t} p(v_{tn}|\boldsymbol{\pi}_{z_{tn}})p(z_{tn}|\boldsymbol{\theta}_t)\Big)p(y_t|\boldsymbol{z_t}, \boldsymbol{b})\Big)} \qquad (5.3)
\end{aligned}$$

In order to estimate the parameters and assign topics to documents we should evaluate either the likelihood or posterior functions. But both Eq 5.2 and Eq 5.3 turn out to be intractable as in LDA due to the coupling between the $\boldsymbol{\theta}$ and $\boldsymbol{\pi}$ variables. Thus we cannot use the plain vanilla EM algorithm as in the usual setting and will resort to approximate solutions like Gibbs Sampling or Variational EM algorithms. Since this is a supervised model, there are two parts to the estimation where in the E-step the random variables $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$ are estimated and the regression coefficients $\boldsymbol{b}$ are estimated as part of M-step.

5.1.1 Gibbs Sampling Algorithm for SLDA

In this section we will discuss the Collapsed Gibbs sampling algorithm for learning and inference of SLDA parameters as given by Chang [15]. Collapsed Gibbs sampling estimates values of the topic assignments $\boldsymbol{z}$ by collapsing the random variables $\boldsymbol{\theta}$ and parameters $\boldsymbol{\pi}$. In this derivation, a Gaussian distribution is assumed as a specific case of GLM and thus the probability of generating $\boldsymbol{y}$ is given by:

$$\begin{aligned}
p(y_t|\boldsymbol{z_t}, \boldsymbol{b}, \delta) &\propto \exp(-(y_t - \boldsymbol{b}^T\tilde{\boldsymbol{z}}_{\boldsymbol{t}})^2) \\
&\propto \exp(-2(\boldsymbol{b}^T\tilde{\boldsymbol{z}}_{\boldsymbol{t}})y_t - (\boldsymbol{b}^T\tilde{\boldsymbol{z}}_{\boldsymbol{t}})^2) \\
&\quad \text{where terms that are only dependent on } \boldsymbol{z_t} \text{ have been retained}
\end{aligned}$$

The posterior distribution of $\boldsymbol{z}$ is given by:

$$\begin{aligned} p(\boldsymbol{z}|\boldsymbol{\alpha},\boldsymbol{\beta},\delta,\boldsymbol{v},\boldsymbol{y},\boldsymbol{b}) &\propto p(\boldsymbol{v}|\boldsymbol{z},\boldsymbol{\beta})p(\boldsymbol{z}|\boldsymbol{\alpha})p(\boldsymbol{y}|\boldsymbol{z},\boldsymbol{b},\delta) \\ &\propto \prod_t \frac{B(\boldsymbol{\alpha}+\boldsymbol{n}_t)}{B(\boldsymbol{\alpha})} \prod_k \frac{B(\boldsymbol{\beta}+\boldsymbol{n}_k)}{B(\boldsymbol{\beta})} \exp(-2(\boldsymbol{b}^T\tilde{\boldsymbol{z}}_{\boldsymbol{t}})y_t - (\boldsymbol{b}^T\tilde{\boldsymbol{z}}_{\boldsymbol{t}})^2) \end{aligned}$$

and the above posterior has the extra term from the random variable $\boldsymbol{y}$. Using the above equation we arrive at the following sampling equation for the variable z_{tn} for the t^{th} document and n^{th} term.

$$\begin{aligned} p(z_{tn}|\boldsymbol{\alpha},\boldsymbol{\beta},\delta,\boldsymbol{v}_{\boldsymbol{t}},y_t,\boldsymbol{b},\boldsymbol{z}_{\neg(tn)}) \propto \quad & \frac{B(\boldsymbol{n}_t+\boldsymbol{\alpha})}{B(\boldsymbol{n}^t_{-(t,n)}+\boldsymbol{\alpha})} \frac{B(\boldsymbol{n}_k+\boldsymbol{\beta})}{B(\boldsymbol{n}^k_{-(t,n)}+\boldsymbol{\alpha})} \\ & \exp\Big(\frac{b_k}{N_t}(y_t - \boldsymbol{b}^T\tilde{\boldsymbol{z}}_t^{\neg n}) - (\frac{b_k}{N_t})^2\Big) \end{aligned} \tag{5.4}$$

where $\boldsymbol{z}_{\neg(tn)}$ indicates the topic assignments of all words other than z_{tn} and similarly $\tilde{\boldsymbol{z}}_t^{\neg n}$ is the mean topic distribution over all words with z_{tn}. The second term in the above equation is derived from the Gaussian error assumed between the actual and predicted labels. As in LDA, the topic assignments are used to calculate the counts $\boldsymbol{n}_k$ and $\boldsymbol{n}_t$ which are used to estimate $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$ given below:

$$p(\boldsymbol{\pi}_k|\boldsymbol{v},\boldsymbol{\beta}) \approx Dirichlet(\boldsymbol{\pi}_k;(\boldsymbol{\beta}+\boldsymbol{n}_k)) \tag{5.5}$$

$$p(\boldsymbol{\theta}_t|\boldsymbol{v},\boldsymbol{\alpha}) \approx Dirichlet(\boldsymbol{\theta}_t;(\boldsymbol{\alpha}+\boldsymbol{n}_t)) \tag{5.6}$$

The estimation of $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$ pertains to the E-step and we observe that the coefficients $\boldsymbol{b}$ from previous are used in their calculation. In the M-step, the values of the coefficients are estimated using a simple linear regression model. As in a linear regression model the regression coefficients as given below:

$$\boldsymbol{b} = (\tilde{\boldsymbol{z}}^T\tilde{\boldsymbol{z}})^{-1}\tilde{\boldsymbol{z}}^T\boldsymbol{y} \tag{5.7}$$

Now we will discuss the Variational EM algorithm for SLDA where a deterministic approximation of the posterior is dealt with.

Algorithm 7 Collapsed Gibbs sampling algorithm for SLDA by [15]

```
Input: All documents, $\boldsymbol{v}$ and responses $\boldsymbol{y}$
Output: $\boldsymbol{\theta}$ and $\boldsymbol{\pi}$
Initialize $\boldsymbol{z}$ to topics 1 to K using Uniform(1,K)
procedure GIBBS($I$)                                  ▷ $I$ is # of iterations
    for iter = 1 to $I$ do                            ▷ Iterate till burn-in
        for $t = 1$ to $T$ do                         ▷ Iterate over documents
            for $n = 1$ to $N_t$ do                   ▷ Iterate over words
                Sample $z_{tn}$ using Eq 5.4
            end for
        end for
        Update counts $\boldsymbol{n_k}$ and $\boldsymbol{n_t}$
    end for
    Update the variables $\boldsymbol{\theta}$ and $\boldsymbol{\pi}$ using Eqs 5.5 and 5.6
    Update the regression coefficients $\boldsymbol{b}$ using Eq 5.7
end procedure
```

5.1.2 Variational EM algorithm for SLDA

Since the posterior 5.3 cannot be evaluated directly we will again assume a surrogate distribution q as an estimate to the posterior. q is defined over a family of distributions that arrive from a simplified version of SLDA model and we aim to decrease the Kullback Leibler Divergence between the actual posterior p and the surrogate q. The simplified SLDA graphical model is shown in Figure 5.2 and we observe that the variables related to supervised setting are dropped as $\boldsymbol{b}$ can be dealt as a parameter and to untie the coupling between $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$. The new distribution q is modeled by parameters $\boldsymbol{\gamma}$ and $\boldsymbol{\phi}$ which are Dirichlet prior and Multinomial distributions respectively and q is given by:

$$q(\boldsymbol{\theta}_t, \boldsymbol{z}_t|\boldsymbol{\gamma}_t, \boldsymbol{\phi}_t) = q(\boldsymbol{\theta}_t|\boldsymbol{\gamma}_t)\prod_{n=1}^{N_t} q(z_{tn}|\boldsymbol{\phi}_{tn})$$

It is interesting to note that from the above definition, q is independent of the response variables. As discussed in Section 4.1.3, minimizing KLdivergence is equivalent to increasing the expected lower bound $\mathcal{L}$ given by:

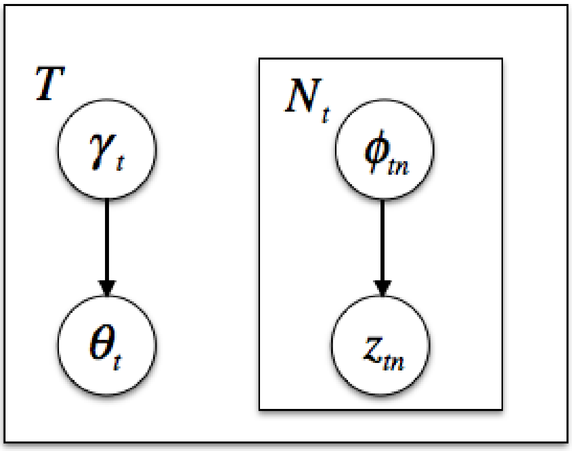


Figure 5.2: A simplified graphical model for SLDA to approximate the actual posterior using simple surrogate family of distributions.

$$\begin{aligned}\mathcal{L}(\boldsymbol{\gamma}, \boldsymbol{\phi}; \boldsymbol{\alpha}, \boldsymbol{\pi}) &= E_q[\log(p(\boldsymbol{\theta}|\boldsymbol{\alpha}))] + E_q[\log(p(\boldsymbol{z}|\boldsymbol{\theta}))] + E_q[\log(p(\boldsymbol{v}|\boldsymbol{z}, \boldsymbol{\pi}))] \\ &\quad -E_q[q(\boldsymbol{\theta})] - E_q[q(\boldsymbol{z})] + E_q[\log p(\boldsymbol{y}|\boldsymbol{z}, \boldsymbol{b})] \qquad (5.8)\end{aligned}$$

The difference between the Elbo formulae of LDA and SLDA is the last term that included the response probability. Since all other terms remain the same we expand the last term here.

$$\begin{aligned}E_q[\log p(y_t|\boldsymbol{z_t}, \boldsymbol{b})] &= E_q\Big[\log(\frac{1}{\sqrt{2\pi\delta}}) + \frac{-(y_t - \boldsymbol{b}^T\tilde{\boldsymbol{z}}_{\boldsymbol{t}})^2}{2\delta}\Big] \\ &= E_q[-\frac{y_t}{2\delta}] + E_q[\frac{y_t\boldsymbol{b}^T\tilde{\boldsymbol{z}}_{\boldsymbol{t}}}{\delta}] + E_q[-\frac{\boldsymbol{b}^T\tilde{\boldsymbol{z}}_{\boldsymbol{t}}\tilde{\boldsymbol{z}}_{\boldsymbol{t}}^T\boldsymbol{b}}{2\delta}] \\ &= \frac{1}{\delta}\Big(\boldsymbol{b}^T E_q[\tilde{\boldsymbol{z}}_{\boldsymbol{t}}]y - E_q[\boldsymbol{b}^T\tilde{\boldsymbol{z}}_{\boldsymbol{t}}\tilde{\boldsymbol{z}}_{\boldsymbol{t}}^T\boldsymbol{b}]\Big) \qquad (5.9)\end{aligned}$$

where $E_q[\tilde{\boldsymbol{z}}_{\boldsymbol{t}}] = \dfrac{1}{N}\sum_n \boldsymbol{\phi_{tn}}$ is evaluated as a mean of all the topic assignment probabilities in a document t. The term $E[\tilde{\boldsymbol{z}}_{\boldsymbol{t}}\tilde{\boldsymbol{z}}_{\boldsymbol{t}}{}^T]$ is needed in both the E and M steps and is given by:

$$E[\tilde{\boldsymbol{z}}_{\boldsymbol{t}}\tilde{\boldsymbol{z}}_{\boldsymbol{t}}{}^T] = \frac{1}{N_t^2}\Big(\sum_n \sum_{m \neq n} \boldsymbol{\phi_n}\boldsymbol{\phi_m}{}^T + \sum_n \mathrm{diag}\{\boldsymbol{\phi_n}\}\Big) \tag{5.10}$$

Following the procedure discussed in Variational EM for LDA model, we now derive the formulae to estimate $\boldsymbol{\phi}$ and $\boldsymbol{\gamma}$. ϕ_{tnk} is calculated by optimizing:

$$\begin{aligned}\underbrace{\text{maximize}}_{\phi^*_{tnk}} \phi_{tnk}(\Psi(\gamma_{tk}) - \Psi(\textstyle\sum \gamma_{tk})) + \phi_{tnk}\log(\pi_{k,v_{tn}}) - \phi_{tnk}\log(\phi_{tnk}) \\ + \frac{y_t \boldsymbol{b}^T E[\tilde{\boldsymbol{z}}_{\boldsymbol{t}}]}{\delta} - \frac{\boldsymbol{b}^T E[\tilde{\boldsymbol{z}}_{\boldsymbol{t}}\tilde{\boldsymbol{z}}_{\boldsymbol{t}}{}^T]\boldsymbol{b}}{2\delta} + \lambda_n(\sum_k \phi_{tnk} - 1)\end{aligned} \tag{5.11}$$

Differentiating the above equation with ϕ_{tnk} and equating it to zero we obtain the optimal value as:

$$(\Psi(\gamma_{tk}) - \Psi(\textstyle\sum \gamma_{tk})) + \log(\pi_{k,v_{tn}}) - \log(\phi_{tnk}) + 1 + (\dfrac{y_t}{N\delta})b_k - \dfrac{\partial}{\partial \phi_{tnk}}\Big\{\dfrac{\boldsymbol{b}^T E[\tilde{\boldsymbol{z}}_{\boldsymbol{t}}\tilde{\boldsymbol{z}}_{\boldsymbol{t}}{}^T]\boldsymbol{b}}{2\delta}\Big\} = 0$$

$$\Longrightarrow$$

$$\phi^*_{tnk} \propto \exp\Big\{(\Psi(\gamma_{tk}) - \Psi(\textstyle\sum \gamma_{tk})) + \log(\pi_{k,v_{tn}}) + (\dfrac{y_t}{N\delta})b_k - \dfrac{1}{2N_t^2\delta}[2(\boldsymbol{b}^T\boldsymbol{\phi_{\neg tn}})\boldsymbol{b} + (\boldsymbol{b}^T\boldsymbol{b})]\Big\} \tag{5.12}$$

If we observe the above estimate of ϕ_{tnk} with that of LDA's estimate in Eq 4.10, there is new term that contains $\boldsymbol{\phi_{\neg tn}}$ which is the mean vector of Multinomials over all words except for the current word (t, n). This makes SLDA a bit slower than LDA in terms of the execution time as the topic assignments of words in each document cannot be done in parallel. And as in LDA, the estimate of $\boldsymbol{\gamma_t}$ remains the same as the one given in Eq 4.11. Once the above parameters are estimated as part of the E-step, as in the Gibbs sampling procedure, the linear predictors $\boldsymbol{b}$ are evaluated in the M-step. Unlike the Gibbs sampling algorithm where the topic assignments

Algorithm 8 Variational Bayes algorithm for SLDA by [12]

```
Input: All Documents $\boldsymbol{v}$, responses $\mathbf{y}$ and Dirichlet Priors $\boldsymbol{\alpha}$, # of EM and E iterations $I$, $J$
Output: Topic distributions $\boldsymbol{\gamma}$ , Term distributions $\boldsymbol{\pi}$ and Linear predictors $\boldsymbol{b}$
Assign equal probabilities to all terms in $\boldsymbol{\pi}_0$
procedure SLDAVBAlgorithm
    while $i < I$ or !$converged$ do                    ▷ Iterate till convergence
        E-Step:
        for each document $t$ do
            Initialize equal values to all topics in $\boldsymbol{\gamma_t}$ as $\gamma_{tk} = \alpha_k + \frac{N_t}{K}$
            while $j < J$ or !$converged$ do
                for each word $n$ do
                    (i) Calculate the Multinomials $\boldsymbol{\phi_{tn}}$ using Eq 5.12
                end for
                (ii) Calculate the Dirichlet priors $\boldsymbol{\gamma_t}$ using Eq 4.11
                $j := j + 1$
            end while
        end for
        M-Step:
        (i) Estimate $\boldsymbol{\pi}_i$ using Eq 4.12
        (ii) Estimate $\boldsymbol{b}$ using Eq 5.13
        $i := i + 1$
    end while
end procedure
```

variable $\tilde{\boldsymbol{z}}_{\boldsymbol{t}}$ are directly used as features, it is a bit complex in Variational inference. The optimal values for $\boldsymbol{b}$ are estimated by maximizing the following function derived from Eq 5.11:

$$\underbrace{\text{maximize}}_{\boldsymbol{b}^*}\Big(\sum_{t=1}^{T}\big(\frac{y_t\boldsymbol{b}^T E[\tilde{\boldsymbol{z}}_{\boldsymbol{t}}]}{\delta} - \frac{\boldsymbol{b}^T E[\tilde{\boldsymbol{z}}_{\boldsymbol{t}}\tilde{\boldsymbol{z}}_{\boldsymbol{t}}{}^T]\boldsymbol{b}}{2\delta}\big)\Big)$$

Differentiating the above function with $\boldsymbol{b}$ and equating to zero we arrive at:

$$\frac{1}{\delta}\Big\{\sum_t E[\tilde{\boldsymbol{z}}_{\boldsymbol{t}}]y_t - \boldsymbol{b}\sum_t E[\tilde{\boldsymbol{z}}_{\boldsymbol{t}}\tilde{\boldsymbol{z}}_{\boldsymbol{t}}{}^T]\Big\} = 0$$

$$\Longrightarrow$$

$$\boldsymbol{b}^* = (E[\tilde{\boldsymbol{z}}_{\boldsymbol{t}}\tilde{\boldsymbol{z}}_{\boldsymbol{t}}{}^T])^{-1}E[\tilde{\boldsymbol{z}}]^T\boldsymbol{y} \tag{5.13}$$

where $E[\tilde{\boldsymbol{z}}_{\boldsymbol{t}}\tilde{\boldsymbol{z}}_{\boldsymbol{t}}{}^T]$ is calculated using Eq 5.10. When we compare this estimate with that of the Gibbs estimate give in Eq 5.7 they are very similar in the sense that they represent the solution of linear regression. But the difference is that the features in this case are the variational Multinomial distributions unlike in Gibbs sampling where the features are the actual empirical topic assignnments. The values of the topic-term Multinomials $\boldsymbol{\pi}$ are estimated using the same equation used in LDA. Also another parameter that can also be estimated is the dispersion or the error variance δ which can also be estimated by differentiating the function. But from our experiments we found that doing so over fits the linear predictors onto the training documents and thus the better approach will be to select δ using cross validation. Same is also the case with the Dirichlet parameter α.

Algorithm 8 consolidates all the steps involved in the Variational inference for SLDA model. It is very similar to the LDA model except for the estimate updates and we also note that in each E-step within each document, there is an extra loop for each word n and this is to indicate that calculation of ϕ_{tn} is dependent on the previous word and cannot be technically executed in parallel. We have used a similar structure as in LDA for implementing SLDA model using a Map reduce framework.

An interesting implementation is that since the log likelihood in SLDA is a combination of unsupervised terms as in LDA and terms that contain supervised responses, we have calculated two likelihoods one that comprises unsupervised terms and the other the supervised ones. Listing B.4 contains the sample code that computes these two likelihoods separately and while checking for convergence the sum of both likelihoods is used. In the Results section we will analyze how these likelihoods evolve after each EM step. The space complexity of the SLDA Variational EM algorithm is the same as LDA but the time complexity is greater than that of LDA by the mere fact that the E step for a word is dependent on rest of the words. The time complexity of SLDA is of the order $O(I * ((T * J * N * K) + (T * N * K + K * V)))$ and the additional term comes from the M step of SLDA where we need to estimate the values $E[\tilde{\boldsymbol{z}}_{\boldsymbol{t}}\tilde{\boldsymbol{z}}_{\boldsymbol{t}}{}^T]$ which is of the order $O(T * N * K)$ where I, J, T, N, K and V are the # of EM iterations, # of E iterations, # of documents, average # of words per document, # of topics and vocabulary size respectively.

Inference and Prediction on Unseen Documents:

What we have seen so far is learning or model training where the needed distributions and linear predictors are learnt from a set of *training* documents. SLDA inference on new unseen documents is performed exactly like in LDA where just the E step shown in Algorithm 4 is performed on the test documents. We cannot use the E step in Algorithm 8 as it needs the response labels for the documents which are not available for *testing* documents. Unlike LDA where it is just a dimension reduction technique, SLDA is not only a dimension reduction technique but also a linear regression model. Once the topic features , $\boldsymbol{\gamma}$ are extracted from a test document the response variable y is predicted as $y = \boldsymbol{\gamma}^T\boldsymbol{b}$. Even though SLDA can be used to predict the continuous response variable, since it is a linear model it does not capture the non linear dependency between the topics and the response variables. So we also test SLDA features by using other regression algorithms like Support Vector Regression with different kernels.

5.1.3 Experiments and Results

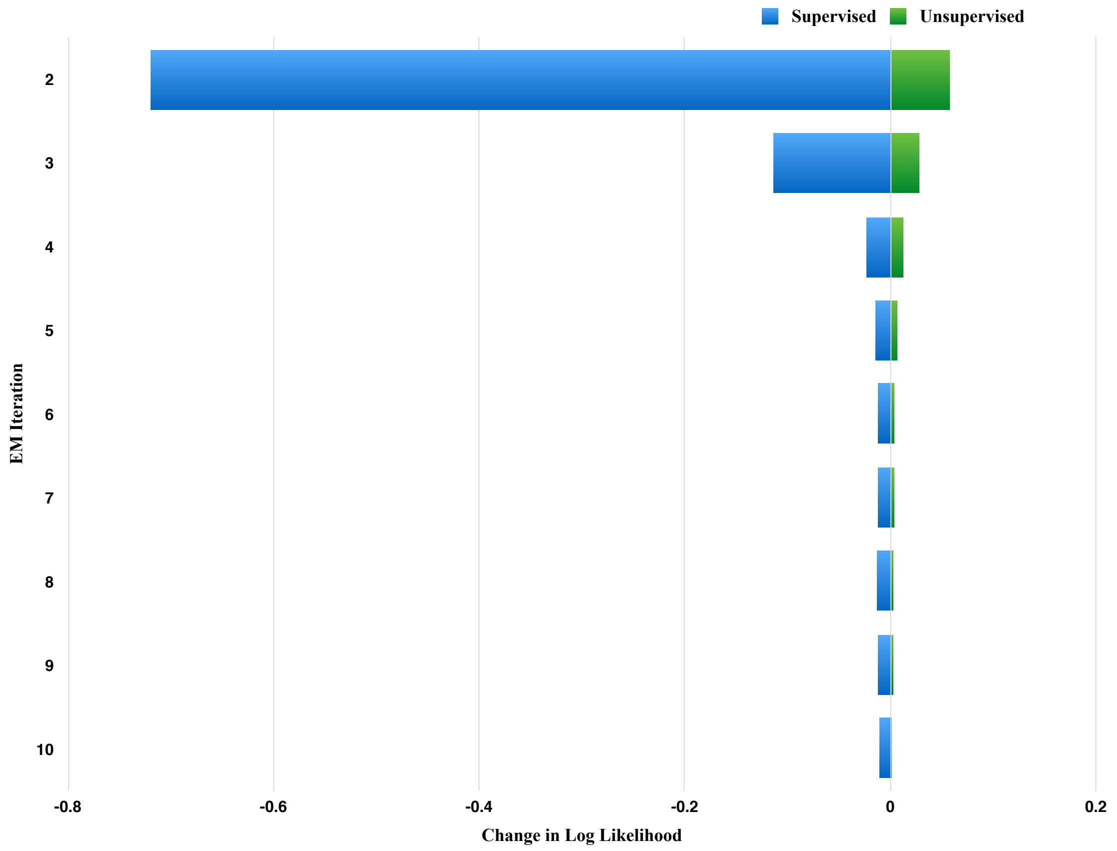


Figure 5.3: Changes in likelihood for supervised and unsupervised terms after each EM Iteration in SLDA model for Landmark features

Comparison of SLDA vs LDA:

We have used AVEC12 dataset to test the efficacy of SLDA models. We have picked up the Audio, LM and LBPTop features to test SLDA model because they have given

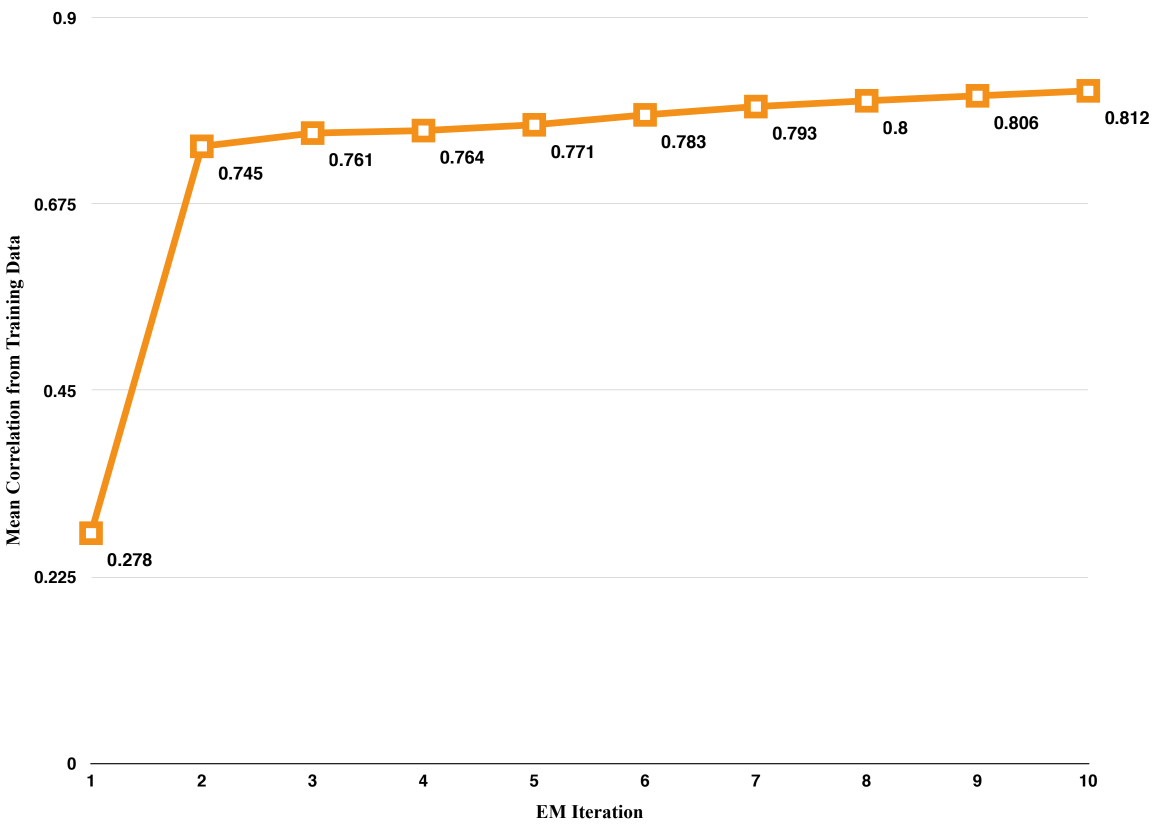
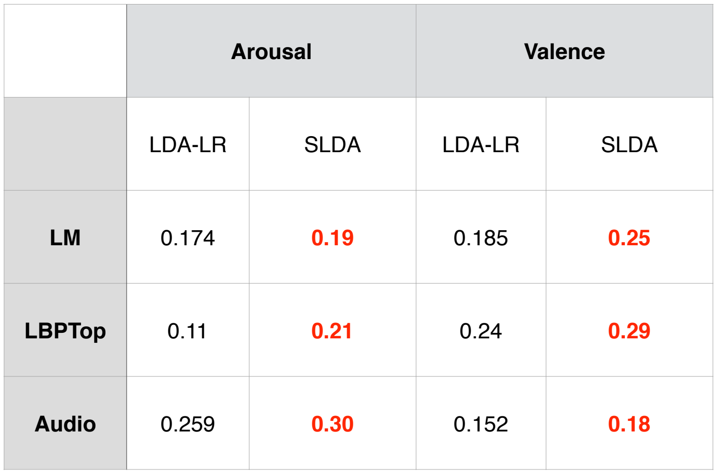


Figure 5.4: Plot of Mean Correlations on Training Data after every EM Iteration in SLDA model. The response variable here is Valence.

the most stable results across various regressors when tested with LDA models. Before moving to the prediction capabilities of the SLDA model it is essential to understand how the EM algorithm affects the prediction capability of the model. Figure 5.3 shows the plot of the changes in the joint likelihood or ELBO value given in Eq 5.8 for SLDA model trained on the LM features. In this figure we separately calculated and plotted the likelihood from the supervised and unsupervised terms in Eq 5.8 and thus the sum of changes tends towards zero. It is interesting to observe that the unsupervised likelihood changes converge to zero sooner than the supervised likelihood changes. E.g. in Figure 5.3, there is not significant change in the unsupervised likelihood after 7-th iteration whereas the supervised likelihood does change for the rest of the iterations.

Figure 5.5: The mean cross correlations on AVEC12 dataset using SLDA and LDA-Linear topic models



	Arousal		Valence	
	LDA-LR	SLDA	LDA-LR	SLDA
LM	0.174	**0.19**	0.185	**0.25**
LBPTop	0.11	**0.21**	0.24	**0.29**
Audio	0.259	**0.30**	0.152	**0.18**

It is also interesting to see how the mean cross correlations across all training data vary with each EM iteration. Figure 5.4 contains a plot of the mean correlation across all AVEC12 videos for response Valence using SLDA on the LM features. We observe that in SLDA model the training correlations increase with iteration or equivalently, the training error decreases with the iterations. This signifies that the supervised nature of SLDA is able to model both topics and responses as the EM algorithm converges to optimal parameters.

The first evaluation criteria is the comparative performance of SLDA model with LDA in relation to arousal and valence prediction. Since running SLDA model is equivalent to the combination of LDA and Linear regression (LDA-LR), we compare them in Table 5.5. It is evident from the results in Table 5.5 that SLDA model in deed performs better than LDA and this true across the features and emotion dimensions.

It is also interesting to see that the trend that has been observed in the LDA results also stands out from SLDA results, that Audio modality has performed again well for the Arousal dimension whereas video modality has performed better for the Valence dimension.

Our experiments have shown that supervised LDA model is an extremely useful supervised dimension reduction technique and also that it is a promising model for multimodal video based time series data. There has not been much research on supervised dimension reduction techniques for regression modeling whereas supervised models like Fischer's Latent Discriminant Analysis [25] and its extensions are used in classification settings. Since SLDA model internally behaves like a linear regression, it inherits the pitfalls from it and therefore tends to overfit the topic features to emotions. In order to address overfitting we provide an extension to SLDA called the regularized SLDA (RSLDA) model by assuming hyper prior over the linear predictors that model the Gaussian error. We will discuss the inference and results from RSLDA model in Section 5.2.

5.2 Regularized Supervised Latent Dirichlet Allocation model

Supervised LDA model that has been discussed in the previous section is a supervised generative model that assumes a generalized linear model while modeling the dependency between response and topics. But since (a) it is an iterative learning scheme and the convergence is evaluated in terms of the likelihood and not response prediction, and (b) it is based on linear regression model, it tends to overfit the responses onto the topic features. Figure 5.6 shows the plot of correlations on training and test data on landmark features after every SLDA EM iteration. We observe that the training correlation increases with each EM Iteration and the correlation of test data also increases. But the correlations on test data decrease after few iterations e.g. in Figure 5.6 it starts decreasing from iteration 5. This indicates that from that EM step onwards SLDA is probably starting to over fit the topics onto the training responses. In a typical regression setting over fitting is most often reduced by using regularization. We have facilitated regularization into topic model by adding an extra prior over the parameters of the generalized linear model and specifically over the Gaussian probability density function.

Figure 5.7 graphically presents the Regularized SLDA (RSLDA) model where in addition to all the variables from SLDA model there is an extra Gaussian prior r over the linear predictors $\boldsymbol{b}$. The probability density function of $\boldsymbol{b}$ is thus $p(\boldsymbol{b}|\boldsymbol{r}) \sim \mathcal{N}(0, \boldsymbol{R}^{-1})$ where $\boldsymbol{R}$ is a diagonal matrix with regularization parameters occupying the diagonal values. We will address the regularization parameter as also ridge parameter. The data generation process for RSLDA is the same as SLDA except that the linear predictors are also generated as below:

Generative Process for RSLDA:

1. Draw a Multinomial K-topic distribution $\boldsymbol{\theta} \sim$ Dirichlet($\boldsymbol{\alpha}$)
2. For each of the N words,
   (a) Assign a topic, $z_n \sim$ Multinomial($\boldsymbol{\theta}$) to n^{th} word
   (b) Draw a term $w_n \sim$ Multinomial($\boldsymbol{\pi}_{z_n}$)
3. Draw the linear predictors $\boldsymbol{b} \sim$ Gaussian($0, \boldsymbol{R}^{-1}$)
4. Draw the response variable $y_t \sim$ Gaussian($\tilde{\boldsymbol{z}}_{\boldsymbol{t}}, \boldsymbol{b}, \delta$)

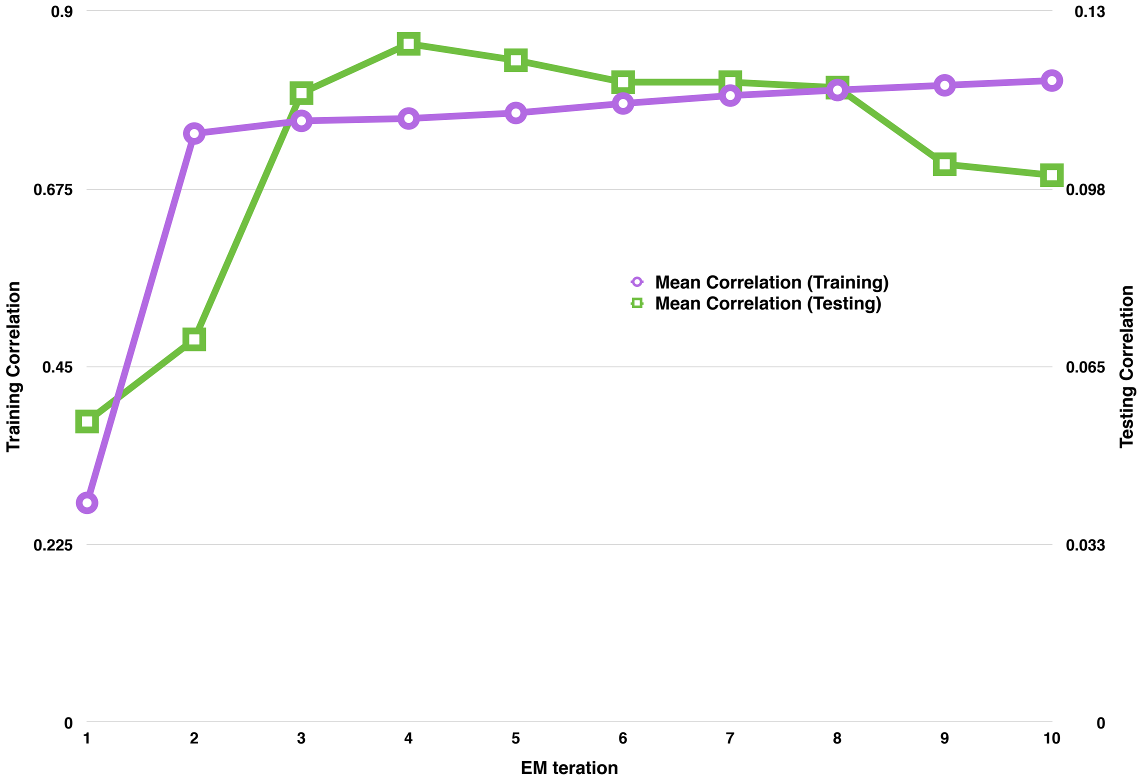


Figure 5.6: Plot of Mean Correlations on Training and Testing Data after every EM Iteration in SLDA model using Landmark features.

The joint distribution of RSLDA model is given by:

$$\begin{aligned} p(\boldsymbol{\theta_t}, \boldsymbol{z_t}, \boldsymbol{v_t}, y_t, \boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{b}, \boldsymbol{r}) &= p(\boldsymbol{\theta_t}|\boldsymbol{\alpha})p(\boldsymbol{z_t}|\boldsymbol{\theta_t})p(\boldsymbol{v_t}|\boldsymbol{z_t}, \boldsymbol{\pi})p(y_t|\boldsymbol{z_t}, \boldsymbol{b}) \\ &= p(\boldsymbol{\theta_t}|\alpha)\Big(\prod_{n=1}^{N_t} p(v_{tn}|\boldsymbol{\pi}_{z_{tn}})p(z_{tn}|\boldsymbol{\theta}_t)\Big)p(y_t|\boldsymbol{z_t}, \boldsymbol{b})p(\boldsymbol{b}|\boldsymbol{r}) \end{aligned}$$

where the additional term is from the prior on the linear predictors $\boldsymbol{b}$. Using a similar Variational approach to inference as in SLDA the ELBO $\mathcal{L}$ is given by:

$$\begin{aligned} \mathcal{L}(\boldsymbol{\gamma}, \boldsymbol{\phi}; \boldsymbol{\alpha}, \boldsymbol{\pi}, \boldsymbol{r}) &= E_q[\log(p(\boldsymbol{\theta}|\boldsymbol{\alpha}))] + E_q[\log(p(\boldsymbol{z}|\boldsymbol{\theta}))] + E_q[\log(p(\boldsymbol{v}|\boldsymbol{z}, \boldsymbol{\pi}))] \\ & -E_q[q(\boldsymbol{\theta})] - E_q[q(\boldsymbol{z})] + E_q[\log p(\boldsymbol{y}|\boldsymbol{z}, \boldsymbol{b})] \\ & +E_q[\log p(\boldsymbol{b}|\boldsymbol{r})] \end{aligned} \tag{5.14}$$

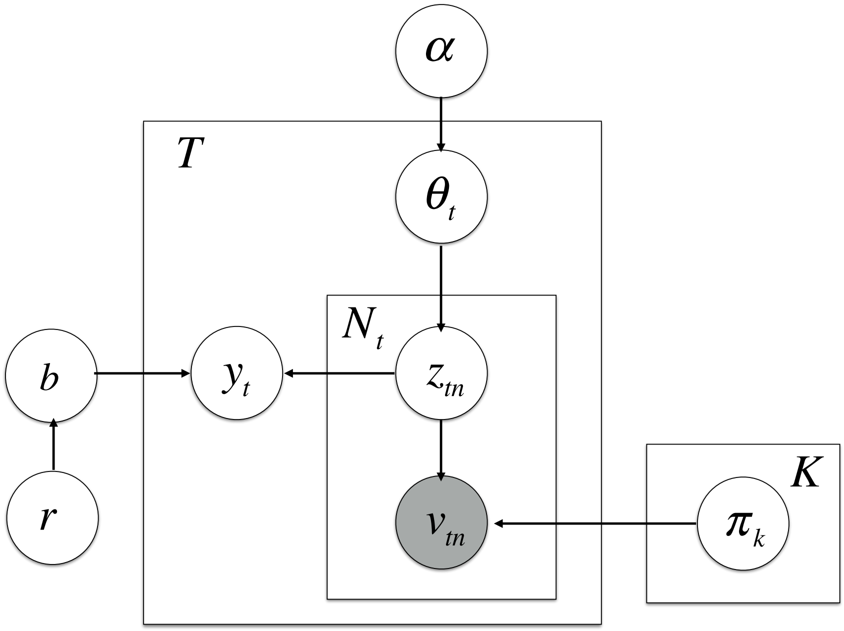


Figure 5.7: Graphical representation of the Regularized Supervised Latent Dirichlet Allocation Model.

The inference on variational parameters $\boldsymbol{\phi}$ and $\boldsymbol{\gamma}$ remain the same as in SLDA and thus regularization does not effect the E step at all. The updates differ for the linear predictors $\boldsymbol{b}$ and thus the M step is a bit different from that in the SLDA algorithm. Let us consider the last two terms from the Eq 5.14 that involve $\boldsymbol{b}$:

$$\begin{aligned} E_q[\log p(\boldsymbol{y}|\boldsymbol{z},\boldsymbol{b})] + E_q[\log p(\boldsymbol{b}|\boldsymbol{r})] &= \frac{1}{\delta}\sum_t \Big(\boldsymbol{b}^T E_q[\tilde{\boldsymbol{z}}_{\boldsymbol{t}}]y - E_q[\boldsymbol{b}^T \tilde{\boldsymbol{z}}_{\boldsymbol{t}} \tilde{\boldsymbol{z}}_{\boldsymbol{t}}{}^T \boldsymbol{b}]\Big) \\ &\quad + E_q\Big[\log(\frac{\boldsymbol{R}}{\sqrt{2\Pi}}) - \boldsymbol{b}^T\boldsymbol{b}\boldsymbol{R}\Big] \end{aligned}$$

By maximizing the above function with respect to $\boldsymbol{b}$ and thus differentiating and equaling it to zero we have:

$$\frac{1}{\delta}\Big\{\sum_t E[\tilde{\boldsymbol{z}}_{\boldsymbol{t}}]y_t - \boldsymbol{b}\sum_t E[\tilde{\boldsymbol{z}}_{\boldsymbol{t}}\tilde{\boldsymbol{z}}_{\boldsymbol{t}}{}^T]\Big\} + \Big\{E[-\boldsymbol{R}\boldsymbol{b}]\Big\} = 0$$

$$\Longrightarrow$$

$$\boldsymbol{b}^* = \frac{1}{\delta}\left(\frac{1}{\delta}E[\tilde{\boldsymbol{z}}_{\boldsymbol{t}}\tilde{\boldsymbol{z}}_{\boldsymbol{t}}^T] + \boldsymbol{R}\right)^{-1} E[\tilde{\boldsymbol{z}}]^T\boldsymbol{y} \tag{5.15}$$

By assuming a symmetric ridge over all topic features we can write $R = rI$ where r is a scalar ridge parameter and I is a K-dimentional identity matrix. We find that the formula to calculate regression coefficients in Eq 5.15 is similar to that of ridge regression or Tikhonov regularization [58].

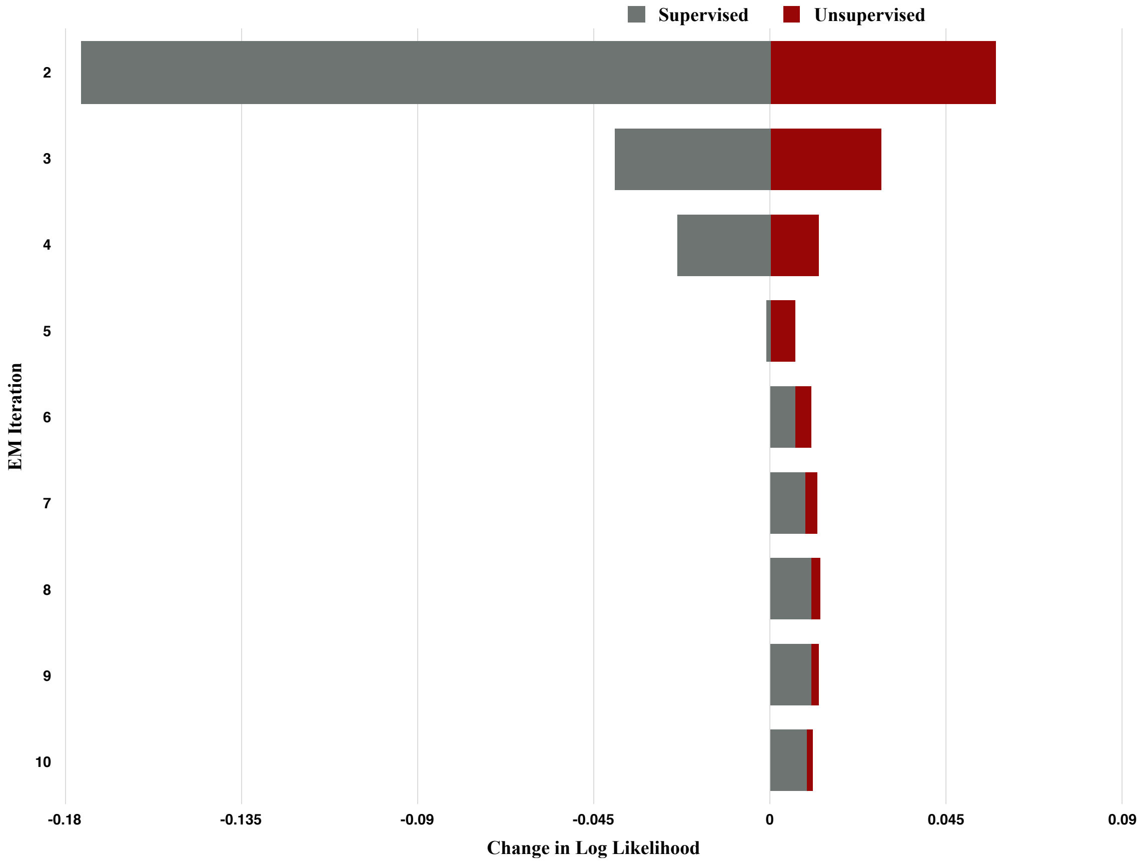


Figure 5.8: Changes in likelihood for supervised and unsupervised terms after each EM Iteration in RSLDA model for Landmark features

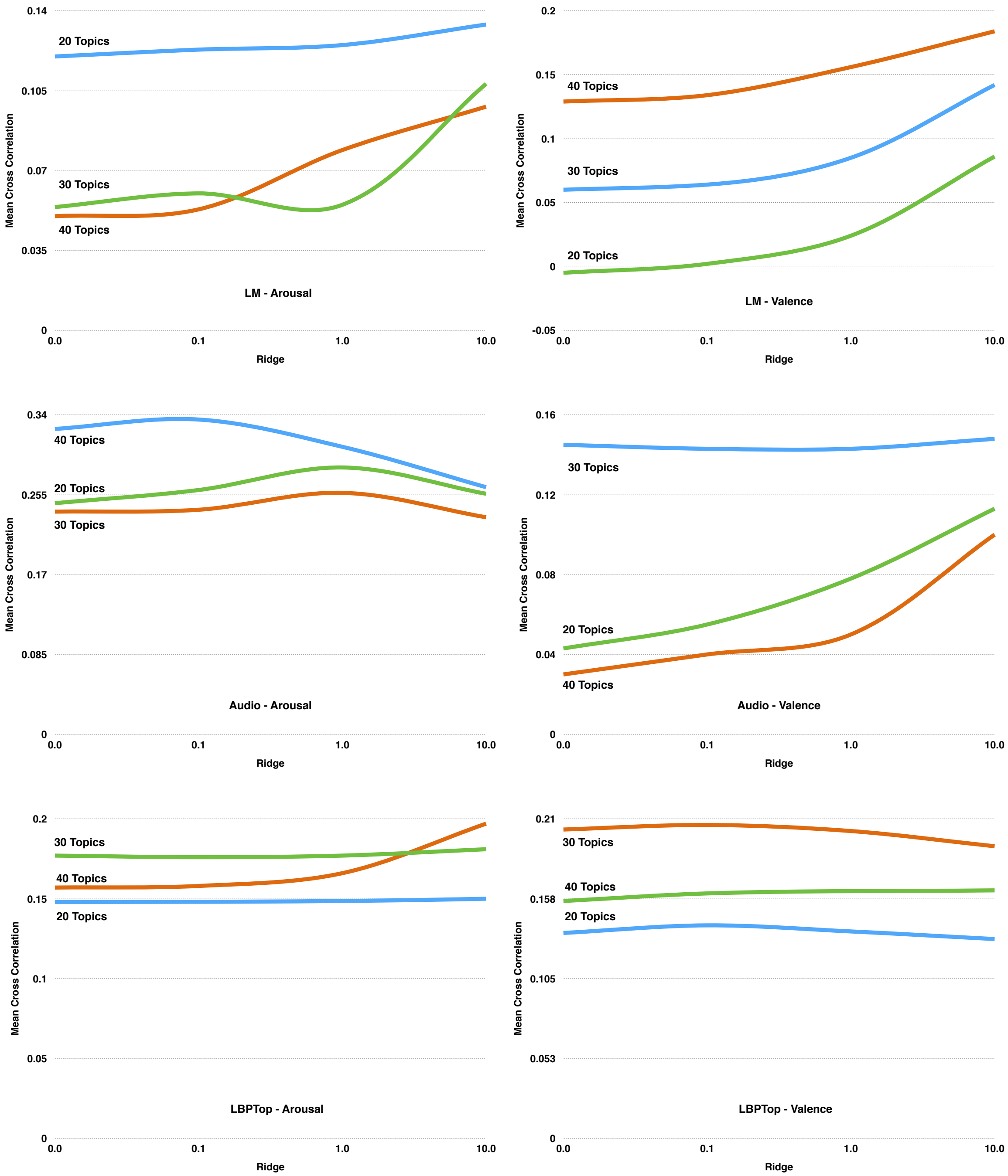


Figure 5.9: Cross Validation results for RSLDA based topic features on LM, Audio and LBPTop features using Linear regression. The plots show the effect of ridge parameters on the performance.

5.2.1 Experiments and Results

Effect of Parameters

The parameters for the RSLDA model, # of topics, α values and the ridge values and we selected them using cross validation. Figure 5.9 contains plots from cross validation over the ridge parameter for different topics generated from LM, Audio and LBPTop features. The results with ridge value 0.0 correspond to plain vanilla SLDA model. These plots indicate that for all features, independent of the # of topics, the best cross validation results comes from a non zero ridge value. This is very promising and exactly replicates the scenario of a regularized regression model.

Comparison of RSLDA vs SLDA

The goal of this evaluation is to find if regularization improves the performance of the SLDA model and if RSLDA model provided a more generalized model. In Figure 5.8 we have plotted the changes in log likelihood from the unsupervised and supervised terms of RSLDA. Unlike the changes in likelihood values of supervised terms in SLDA where the change tends to zero and thereby overfits the topics, in RSLDA we observe that even though the total likelihood change converges to zero, the supervised likelihood changes do not. E.g. In Figure 5.8, the supervised likelihood changes converge to zero until Iteration 5 and then in fact increases again. To understand how this effects the training and testing errors, we have plotted the mean correlations across all training and development videos in Figure 5.10. In this figure we observe that the Training correlations increase and then decrease but the corresponding Testing correlation keeps increasing until convergence. It is interesting to not that the exact point of change in the Training correlation occurs at Iteration 6 which corresponds to the increase in the likelihood change.

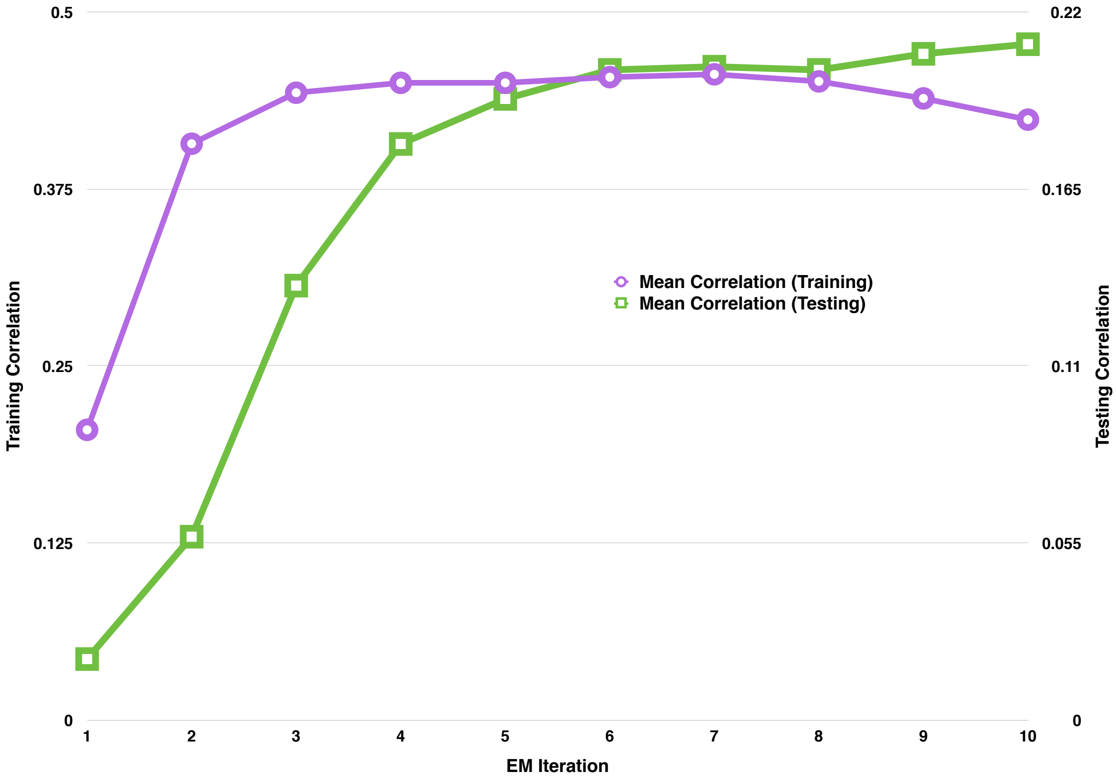
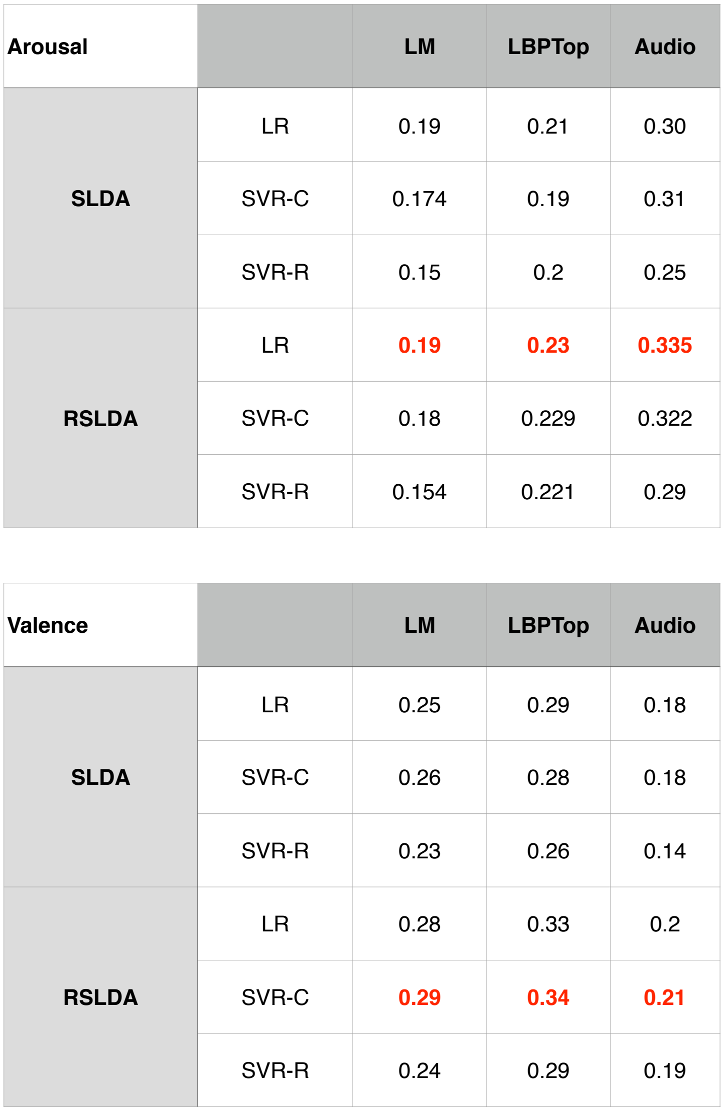


Figure 5.10: Plot of Mean Correlations on Training and Testing Data after every EM Iteration in RSLDA model using Landmark features.

Comparing Figures 5.6 and 5.10, we clearly observe the benefits of RSLDA model over SLDA. Instead of overfitting the topics, it builds a more generalized model.

We have evaluated RSLDA model over all the features using AVEC12 development dataset and compared its performance against the SLDA model without regularization. Table 5.11 contains results from regularized and unregularized SLDA models from three different regressors. The regressor LR in Table 5.11 does not mean that we have separately applied Linear regression but it means that we have directly used the linear predictors generated by SLDA and RSLDA models. RSLDA gave the

Figure 5.11: Cross correlation using RSLDA and SLDA models on AVEC12 development set videos.



Arousal		LM	LBPTop	Audio
SLDA	LR	0.19	0.21	0.30
	SVR-C	0.174	0.19	0.31
	SVR-R	0.15	0.2	0.25
RSLDA	LR	**0.19**	**0.23**	**0.335**
	SVR-C	0.18	0.229	0.322
	SVR-R	0.154	0.221	0.29

Valence		LM	LBPTop	Audio
SLDA	LR	0.25	0.29	0.18
	SVR-C	0.26	0.28	0.18
	SVR-R	0.23	0.26	0.14
RSLDA	LR	0.28	0.33	0.2
	SVR-C	**0.29**	**0.34**	**0.21**
	SVR-R	0.24	0.29	0.19

best performance over all features and Support Vector Regression with Cosine kernel performed best among all the models.

Comparison of RSLDA vs Supervised Dimension Reduction models

In Sections 4.1.4 and 4.2.2 we have compared performance of LDA and DGMM against unsupervised dimension reduction techniques like PCA and KPCA. Since RSLDA is a supervised technique it is prudent that we compare RSLDA model with supervised dimension reduction techniques. We have considered two supervised dimension reduction techniques, 1) Correlation based Supervised Feature Selection (CSF) and 2) Supervised PCA (SPCA). In CSF model we considered each dimension of the feature sets LM, Audio and LBPTop separately and selected the features that are positively correlated to the response or label space (arousal and valence). Supervised PCA model was first proposed by Bair et al. [9] wherein the following methodology is used to reduce the dimensions by including the labels in the process:

1. For each dimension of the feature space, build univariate regression models over the responses

2. Consider a subset of features whose regression coefficients are greater than a threshold t

3. Perform PCA on the subset of features

where we have used cross validation to select the threshold values t by modeling arousal and valence separately.

We have plotted the mean cross correlations for all features using RSLDA, CSF and SPCA models from AVEC12 dataset in Figure 5.12. The mean correlations have been averaged across three regressors, the linear regression, support vector regression with Rbf and with cosine kernels. The error bars indicate the standard deviations of correlations from all three regressor. The results form Plot 5.12 show that RSLDA model performs better than both CSF and SPCA models. The results also prove that

RSLDA based topic features are stable across the board whereas the CSF and SPCA based features have huge variance and indicate instability or noise in the feature space.

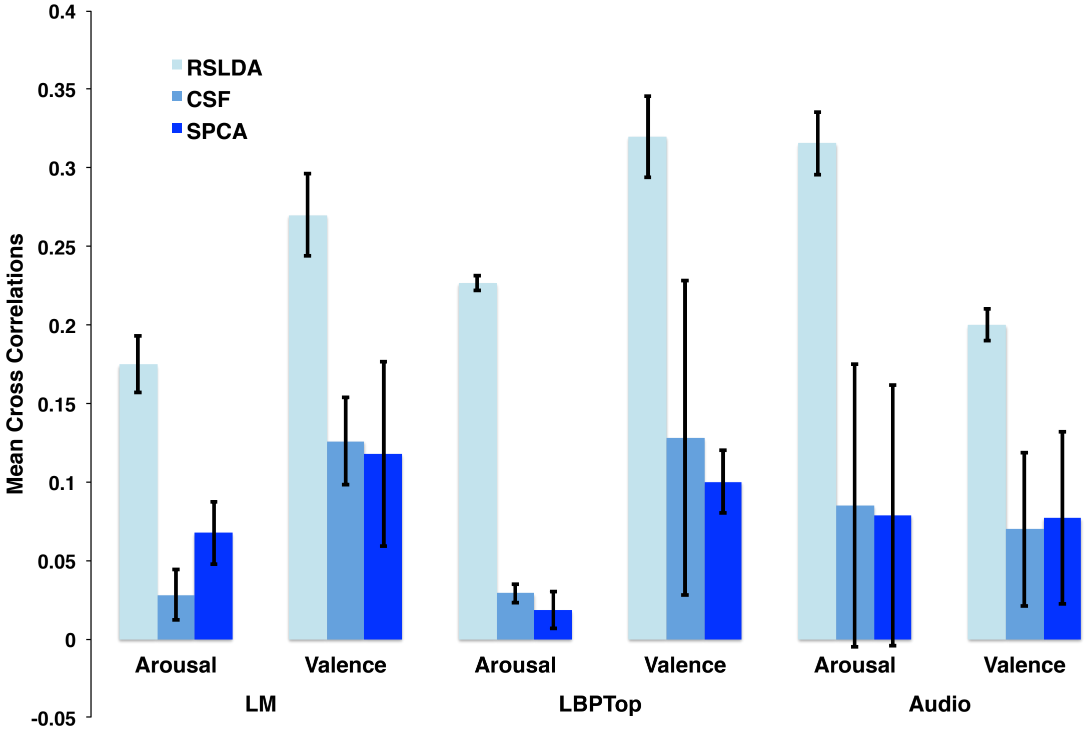


Figure 5.12: Plot of Mean Correlations on AVEC12 development data using RSLDA, CSF and SPCA models. The results are averaged across three regressors.

Another observation we would like to make at this juncture is that the results in Sections 4.1.4, 5.1.3 and 5.2.1 have shown that the Cosine Kernels have performed very well in comparison to RBF kernels especially for the topic features. The reason behind this is that cosine distance helps capture distances between density functions and since topic features are Multinomial probability distributions, cosine kernels are able to group features accurately. To generalize the performance of RSLDA model, we have also tested it using AVEC14 dataset with 50 training videos and 50 development videos. Figure 5.13 contains the mean cross correlations across three regressors using video and audio features. For this dataset we have used LGBP features described in Section 3.1 as the video features and as with AVEC12 results, we can observe that for RSLDA has again outperformed both CSF and SPCA models.

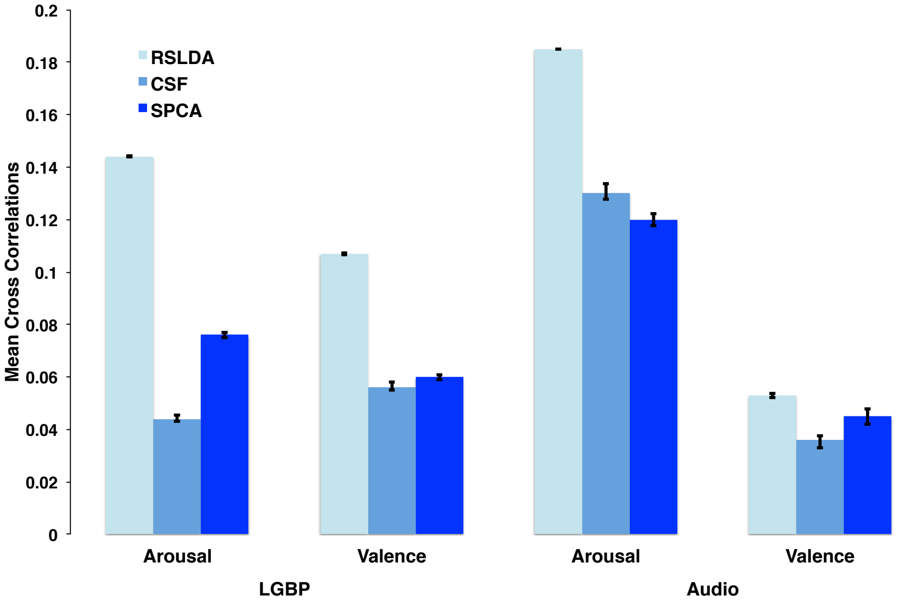


Figure 5.13: Plot of Mean Correlations on AVEC14 development data using RSLDA, CSF and SPCA models. The results are averaged across three regressors.

Multimodal Fusion

Until now we have tested topic features from different models on each base feature individually. But we observe that certain modalities perform better for certain emotions e.g. audio modalities performs better on arousal and video performs better on valence dimension. To make use of the capabilities of both modalities we have tested

Table 5.1: Multimodal fusion results from RSLDA model. V1 - LM, V2 - LBPTop and A1 - Audio features

Dimension	Arousal			Valence		
	LR	SVR-C	SVR-R	LR	SVR-C	SVR-R
V1	0.19	0.18	0.154	0.28	0.29	0.24
V2	0.23	0.229	0.221	0.33	0.34	0.29
A1	0.335	0.322	0.29	0.2	0.21	0.19
A1V1	0.27	0.23	0.24	0.27	0.28	0.25
A1V2	0.34	0.32	0.31	0.37	0.38	0.32
A1V1V2	**0.367**	0.364	0.273	**0.39**	0.38	0.3

RSLDA models by performing a multimodal fusion of audio and video features. By fusion we mean that we have concatenated the features from different modalities and trained the regression models one new combine features. Table 5.1 contains the results from multimodal fusion of RSLDA features on AVEC12 dataset. It can be seen that the combined results of two modalities is better than the performance of the lowest modality. E.g. the average correlation from A1V1 is 0.24 which is greater than the performance of V1 with average correlation 0.174 for arousal prediction. The combination of Audio and LBPTop features has boosted the performance of both arousal and valence. As pointed out earlier, Audio and LBPTop performed better on arousal and valence respectively but their fusion boosted prediction of both dimensions.

5.3 Conclusions

In this chapter, we have explored learning techniques that perform dimension reduction conjointly with supervised learning. Supervised LDA models provide a natural extension to unsupervised topic models by including the linear predictors for regression within the model structure. Since we have considered continuous emotion recognition we have used a SLDA with a Gaussian distribution error and showed that they perform better than LDA models. Similar to a linear regression model, SLDA overfits topics from training data onto the the emotions and this restricts the generalizability of SLDA model. We thus proposed a regularized extension to SLDA model by including hyper priors on the linear predictors. The new RSLDA models have generalized very well in comparison to SLDA and we have also evaluated RSLDA against some of the supervised dimension reduction techniques like SPCA. Topic models have shown a lot of promise in reducing the dimensions as well as improving the performance and at the same time by maintaining stability of the features. Unlike other dimension reduction techniques or feature selection models, RSLDA based topic features gave a stable performance across regressors and this is very promising. Since certain modalities tend to model certain emotions better, we used feature fusion to concatenate features from different modalities. This improved the performance much more and the final results of RSLDA on arousal and valence emotions have increased in comparison to all the results from individual modalities.

Throughout chapters 4 and 5 we have compared the results from different topic models against other dimension reduction techniques. Even though the focus of this research is to evaluate the performance of topic models as dimension reduction models, we would like to discuss how these results compare to the state of the art models on AVEC12 datasets. Nicole et al. [46] obtained the best results on the AVEC12 dataset with mean cross correlations of 0.35 and 0.644 for valence and arousal respectively. They have used a fusion of appearance based AAM [18] features and audio features and have accounted for the *lag* between predictions and actual labels. In this work we have not concentrated on the lag between the predictions and the actual labels as our aim has not been to maximize the results on a particular dataset.

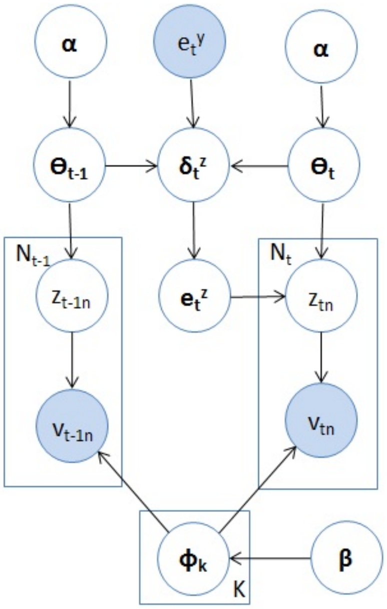
In Chapters 4 and 5 we have discussed the application of topic models to emotion recognition and we have assumed image features to be independent of each other. But while building models for applications like emotion change detection, we cannot make these assumptions. Thus in Chapter 6, we have proposed two temporal topic models to perform change prediction. We have compared them with non temporal topic models and PCA models and have shown that temporal models in fact perform very well.

Chapter 6

TEMPORAL TOPIC MODELS

In Chapters 4 and 5 we discussed topic models in continuous and supervised settings and have always assumed each multimodal document to be independent of the other. This assumption is very restrictive in some applications like emotion change detection where change can be modeled only by considering pair of documents. To handle such scenarios, in this chapter we discuss two new topic models Adaptive Temporal Topic Model (ATTM) and Supervised LDA for Change Detection (SLDACD) that factor in the temporal dependencies between consecutive documents. We specifically created these models to predict changes in emotions rather than the actual emotion values. In applications like mental healthcare and criminal investigations, it is more important to know when the changes in emotion occurs rather than the actual emotion that is displayed.

The premise for work in this chapter is that changes in topic distributions of video frames reflect changes in emotions and vice versa. But existing topic models LDA and SLDA are non-temporal models and so the previous emotion or topic distribution do not effect the topic extraction process. So two temporal topic models have been proposed in this chapter one that models threshold based changes and the other that models Cusum based changes. The first model is called the Adaptive Temporal Topic model (ATTM) and the second one is the Supervised LDA for Change Detection (SLDACD) model which are discussed below.



t : Current document
N_t : # of words in document *t*
K : # of Topics
$\mathbf{z}_t$: n_t-vector of topic assignments to all words in *t*
$\mathbf{v}_t$: n_t-vector of all words that belong to *t*
$\boldsymbol{\phi}_k$: Multinomial distribution of terms per topic
$\boldsymbol{\Theta}_t$: Multinomial distribution of topics per document *t*
$\boldsymbol{\alpha}$: Dirichlet prior vector of $\boldsymbol{\phi}_k$
$\boldsymbol{\beta}$: Dirichlet prior vector $\boldsymbol{\Theta}_t$
e_t^y : Emotion Change Indicator (1/0) for document *t*
$\boldsymbol{e}_t^z$: K-vector of Binomial variables for K topics in *t*
$\boldsymbol{\delta}_t^z$: K-vector of Bernoulli variables for K topics in *t*

Figure 6.1: Graphical model for ATTM model with explanation of notation used

6.1 Adaptive Temporal Topic Model (ATTM)

Adaptive Temporal Topic Model (ATTM) extracts topics from audio/video documents by adapting them to the changes in the human emotional state at each time step. The presence or absence of a significant change in the emotion defines the temporal dependencies between topics in two adjacent frames. Once the topics are extracted as in other topic model based methods regression models are used to calculate the emotions using the new topic features. These values are thresholded to detect change. Figure 6.1 shows the graphical model of ATTM and the notations used in this model. This topic model assumes that every document is part of a time series and let $t-1$ and t be two adjacent documents in a particular sequence. Most of the variables are part of LDA model and have the same interpretation in ATTM also.

The new variables that have been added in ATTM are e_t^y, $\boldsymbol{\delta_t^z}$ and $\boldsymbol{e_t^z}$. e_t^y is the emotion change indicator and takes a value of 1 when there is significant change in the emotion state y_t or 0 otherwise. $\boldsymbol{\delta_t^z}$ is a vector of K Beta variables assigned to each document where each δ_{tk}^z is the probability of selecting a topic k for the document t. The δ_{tk}^z's depend on the variables e_t^y, θ_{t-1k} and θ_{tk}. Each $\delta_{tk}^z \sim$ Beta(α_{tk}^z,β_{tk}^z) where α_{tk}^z and β_{tk}^z are obtained from θ_{t-1k} and θ_{tk}. This calculation is explained in Section 6.1.2. $\boldsymbol{e_t^z}$ is a vector of bernoulli variables (zeros or ones), where each e_{tk}^z is sampled using the probability δ_{tk}^z. Each e_{tk}^z indicates whether the words in the document t can be sampled from the topic k or not. $\boldsymbol{e_t^z}$ vector will be referred as Topic indicator vector. In this model all the variables are hidden except for $\boldsymbol{v}$, and $\boldsymbol{e^y}$.

As in LDA and SLDA collapsed Gibbs sampling has been used to estimate variables $\boldsymbol{z}$ and $\boldsymbol{e^z}$. The joint distribution of all the variables in ATTM is given by :

$$P(\boldsymbol{v}, \boldsymbol{z}, \boldsymbol{e^z}|\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{e^y}) = P(\boldsymbol{v}|\boldsymbol{z}, \boldsymbol{\beta})P(\boldsymbol{z}|\boldsymbol{e^z}, \boldsymbol{\alpha})P(\boldsymbol{e^z}|\boldsymbol{e^y}) \tag{6.1}$$

In the collapsed Gibbs sampling we use the joint distribution to sample a single variable conditioned on remaining variables. We first sample the topic indicators $\boldsymbol{e^z}$ and then use these to sample the topics $\boldsymbol{z}$. The Gibbs sampling algorithm for ATTM is explained in Algorithm 9. A few details about the sampling equations are given in the following sections. For a detailed derivation of these equations please refer to the Appendix A.

6.1.1 Sampling $\boldsymbol{z}$

Assuming that the $\boldsymbol{e^z}$ vector is given, z_{tn} is sampled for each word v_{tn} in each document t. In ATTM while the number of topics is K, the last topic K is used as a dummy topic and the effective number of topics are only $K-1$. Given a topic indicator $\boldsymbol{e_t^z}$, let $\boldsymbol{k'}$ be a set such that $k \in \boldsymbol{k'}$ if $e_{tk}^z = 1$ i.e $\boldsymbol{k'}$ is the set of topics that are selected for this document. Using $\boldsymbol{k'}$ a new Multinomial distribution $\boldsymbol{\hat{\theta}_t}$ is

Algorithm 9 ATTM Gibbs sampling algorithm

```
Input: $\boldsymbol{v}$ and $\boldsymbol{e^y}$
Initialize all $\boldsymbol{z}$ to topics 1 to K using Uniform(1,K)
Initialize all $\boldsymbol{e^z}$ to 0 or 1 using Bernoulli(0.5)
procedure GIBBS($I$)                                    ▷ $I$ is # of iterations
    for  iter = 1 to $I$ do                             ▷ Iterate till burn-in
        for  $s = 1$ to $S$ do                          ▷ Iterate over sequences
            for $t = 1$ to $T$ do                       ▷ Iterate over documents
                if $t = 1$ then
                    Sample $\boldsymbol{z_1}$ using LDA
                else
                    for $k$ =1 to $K - 1$  do
                        Sample $e^z_{tk}$ using Eqs (6.5) and (6.6)
                    end for
                    $e^z_{tK} = 1$                      ▷ Always include topic K
                    for $n = 1$ to $N_t$ do             ▷ Iterate over words
                        Sample $z_{tn}$ using Eqs (6.2), (6.3) and (6.4)
                    end for
                end if
            end for
        end for
    end for
    Update the parameters $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ using expectations of (4.7) and (4.6 )
end procedure
```

generated which is obtained as: IF $k \in \boldsymbol{k'}$ then $\hat{\theta}_{tk} = \theta_{tk}$, IF $k \notin \boldsymbol{k'}$ then $\hat{\theta}_{tk} = 0$ and IF $k = K$ then $\hat{\theta}_{tk} = \sum_{k \notin \boldsymbol{k'}} \theta_{tk} + \theta_{tK}$. We observe that the new $\boldsymbol{\hat{\theta}_t}$ is still a Multinomial distribution and $\boldsymbol{\hat{\theta}_t} \sim \text{Dir}(\boldsymbol{\alpha^{k'}}, \sum_{k \notin \boldsymbol{k'}} \alpha_k + \alpha_K)$. For the term w which is the n^{th} word in document t, the collapsed Gibbs formula for z_{tn} is given by ($\boldsymbol{\alpha}, \boldsymbol{\beta}$ are ignored for brevity):

$$\begin{aligned} P(z_{tn} = k | \boldsymbol{z}_{\neg(tn)}, \boldsymbol{v}, \boldsymbol{e^z}, \boldsymbol{e^y}) &= \frac{P(\boldsymbol{v}, \boldsymbol{z}, \boldsymbol{e^z}, \boldsymbol{e^y})}{P(\boldsymbol{v}, \boldsymbol{z}_{\neg(tn)}, \boldsymbol{e^z}, \boldsymbol{e^y})} \\ &\propto \frac{P(\boldsymbol{v}|\boldsymbol{z})}{P(\boldsymbol{v}_{\neg(tn)}|\boldsymbol{z}_{\neg(tn)})} \frac{P(\boldsymbol{z}|\boldsymbol{e^z})}{P(\boldsymbol{z}_{\neg(tn)}|\boldsymbol{e^z})} \end{aligned}$$

By expanding and deriving the above two ratios the following sampling equations are obtained:

$$\forall k \neq K, P(z_{tn} = k | e^z_{tk} = 0) \sim$$

$$\frac{n^w_{\neg k} + \beta_w}{\sum\limits_{v \neq w} n^v_k + n^w_{\neg k} + \sum\limits_{v=1}^{V} \beta_v} \frac{1}{n_{\neg t} + \sum\limits_{i=1}^{K} \alpha_i} \tag{6.2}$$

$$\forall k \neq K, P(z_{tv} = k | e^z_{tk} = 1) \sim$$

$$\frac{n^v_{\neg k} + \beta_w}{\sum\limits_{v \neq w} n^v_k + n^w_{\neg k} + \sum\limits_{v=1}^{V} \beta_v} \frac{n^k_{\neg t} + \alpha_k}{n_{\neg t} + \sum\limits_{i=1}^{K} \alpha_i} \tag{6.3}$$

$$\text{If } k = K, P(z_{tv} = K) \sim$$

$$\frac{n^w_{\neg K} + \beta_w}{\sum\limits_{v \neq w} n^v_K + n^w_{\neg K} + \sum\limits_{v=1}^{V} \beta_v} \frac{n^K_{\neg t} + \sum\limits_{k \notin \boldsymbol{k'}} \alpha_k + \alpha_K}{n_{\neg t} + \sum\limits_{k=1}^{K} \alpha_k} \tag{6.4}$$

where $n^w_{\neg k}$ is # of times the term w has been assigned to topic k excluding the current assignment, $n^{\tilde{v}}_k$ is # of times the term $\tilde{v}$ has been assigned to topic k, $n_{\neg t}$ is the total # of terms in document t excluding the current term, $n^k_{\neg t}$ is # of times topic k has

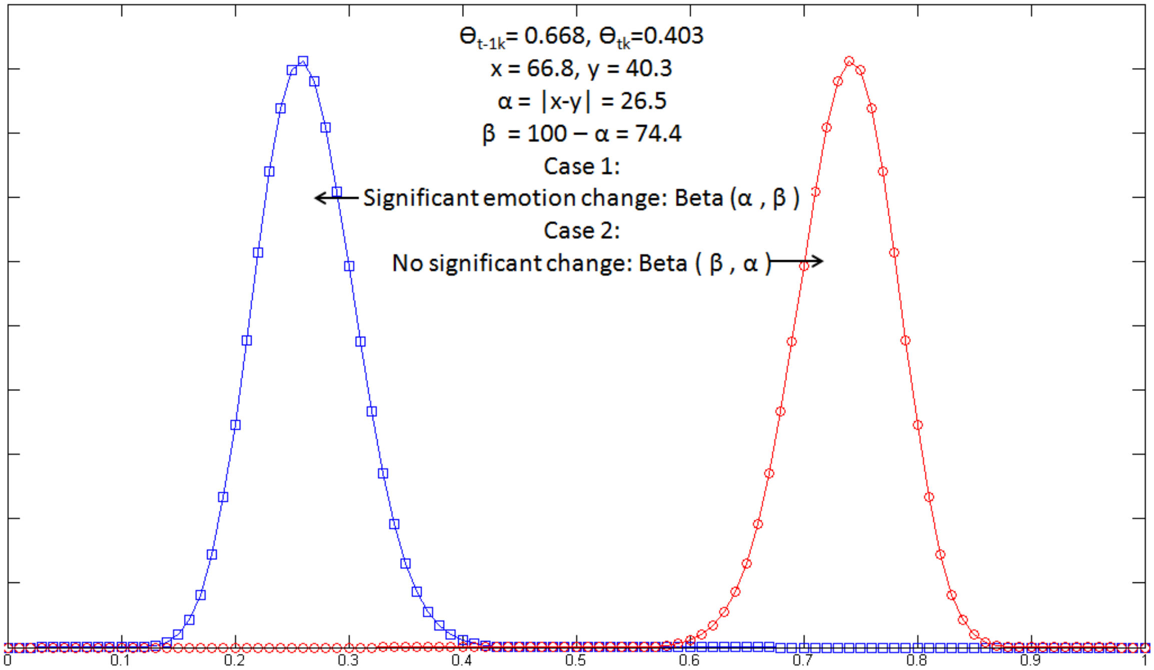


Figure 6.2: Illustration of how change in emotion effects the topic selection. x and y are scaled values of θ_{t-1k} and θ_{tk} respectively.

been assigned to document t excluding the current assignment. The derivations of all these three cases can be found in the Appendix A whose equivalents are derived as Eqs A.13,A.14 and A.17.

6.1.2 Sampling $\boldsymbol{e^z}$

A topic indicator vector of a document indicates whether a topic should be used in the document or not. In ATTM for every document t, a $\boldsymbol{\theta_t}$ distribution is generated using the Dirichlet distribution with $\boldsymbol{\alpha}$ as the parameters. Before sampling the topics from this distribution we want to calculate δ^z_{tk}*'s* which indicate the probability of retaining the topic k in the document. The fundamental principle of ATTM is that, *If there is a significant change, retain topics that are 'different' from the previous document. If not, retain topics that are 'similar' to the previous document.* In the first case for a given topic k, we want δ_{tk}^z to be high when its topic probabilities θ_{t-1k} and θ_{tk} are different. Similarly in the second case the value of δ_{tk}^z has to be high when its topic probabilities are similar. Given the values of θ_{t-1k} and θ_{tk}, we calculate δ_{tk}^z as follows:

Scale θ_{t-1k} and θ_{tk} to the range of [1,100]

Calculate $\alpha_{tk}^z = |\theta_{t-1k} - \theta_{tk}|$ and $\beta_{tk}^z = 100\text{-}\alpha_{tk}^z$

Case 1: Generate $\delta_{tk}^z \sim \text{Beta}(\alpha_{tk}^z,\beta_{tk}^z)$

Case 2: Generate $\delta_{tk}^z \sim \text{Beta}(\beta_{tk}^z,\alpha_{tk}^z)$

The above concept is illustrated in Figure 6.2. In the figure $\theta_{t-1k} = 0.66$ and $\theta_{tk} = 0.4$ and when there is a significant change in emotion we obtain $\delta_{tk}^z \sim 0.25$ which indicate that there are less chances of selecting the topic k because they are very similar. Similarly when there is no significant change in emotion $\delta_{tk}^z \sim 0.75$ that is there is high probability of picking topic k as they are very similar. The Bernoulli variables e_{tk}^z for each topic k are then sampled using δ_{tk}^z as the probability. The vector $\boldsymbol{e_t^z}$ for document t now indicates whether a topic need to be considered for the document. The conjugacy between the Bernoulli variables $\boldsymbol{e_t^z}$ and Beta variables $\boldsymbol{\delta_t^z}$ is used in the derivation of sampling equations for e_{tk}^z which are given by:

$$P(e_{tk}=1) \sim \frac{\prod_{s=0}^{n_t^k}(\alpha_k+s)}{\prod_{s=n_{\neg t}}^{n_t}(\sum_{i=1}^{K}\alpha_i+s)}\frac{\alpha_{tk}^z}{\alpha_{tk}^z+\beta_{tk}^z} \tag{6.5}$$

$$P(e_{tk}=0) \sim \frac{1}{\prod_{s=n_{\neg t}}^{n_t}(\sum_{i=1}^{K}\alpha_i+s)}\frac{\beta_{tk}^z}{\alpha_{tk}^z+\beta_{tk}^z} \tag{6.6}$$

where n_t is the total # of terms in document t, n_t^k is the # of times topic k has been assigned to document t and $n_{\neg t}$ is the total # of terms in document t excluding the words that have been assigned to current topic. The detailed derivations for above equations are given in Section A.2 and the corresponding equations are Eqs A.22, A.23, A.24 and A.25. After calculating $\boldsymbol{e^z}$ and $\boldsymbol{z}$ vectors $\boldsymbol{\theta_t}$ and $\boldsymbol{\phi_k}$ are estimated using 4.7 and 4.6. The $\boldsymbol{\theta_t}$ distributions learned from training documents are used as their

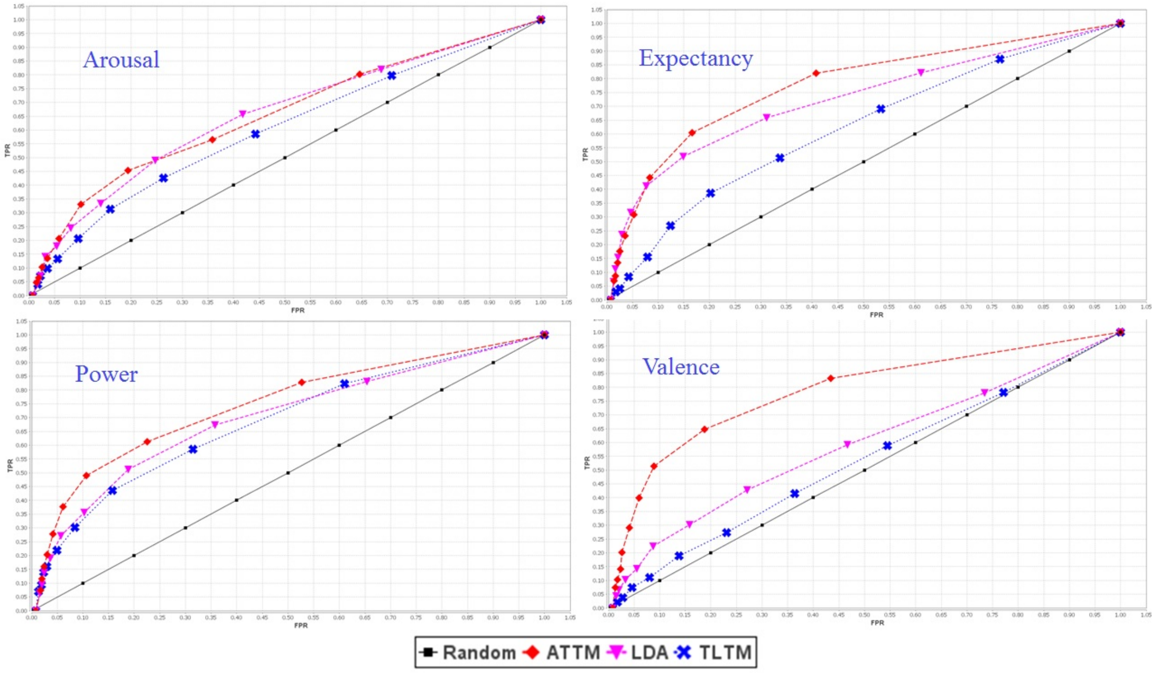


Figure 6.3: ROC curves of ATTM, LDA and TLTM topic models for Arousal, Expectancy, Power and Valence (top-left to bottom-right) using Audio features and Linear Regression for a threshold of 30.0

new features and the $\boldsymbol{\phi_k}$ distributions are used to extract topics from unseen test documents.

6.1.3 Experiments and Results

In continuous emotion recognition, an assumption that is usually made is that the distribution of the emotion remains the same over a period of time. But this assumption necessarily need not hold as the environment may effect the human cognition which may create a drift from the expected behavior. Detecting these change points have a two fold benefit, firstly action can be taken by whenever a change is detected, secondly changes can be used to predict the actual emotion as well by training blocks of cohesive distributions instead of using entire data. In this work we have explored

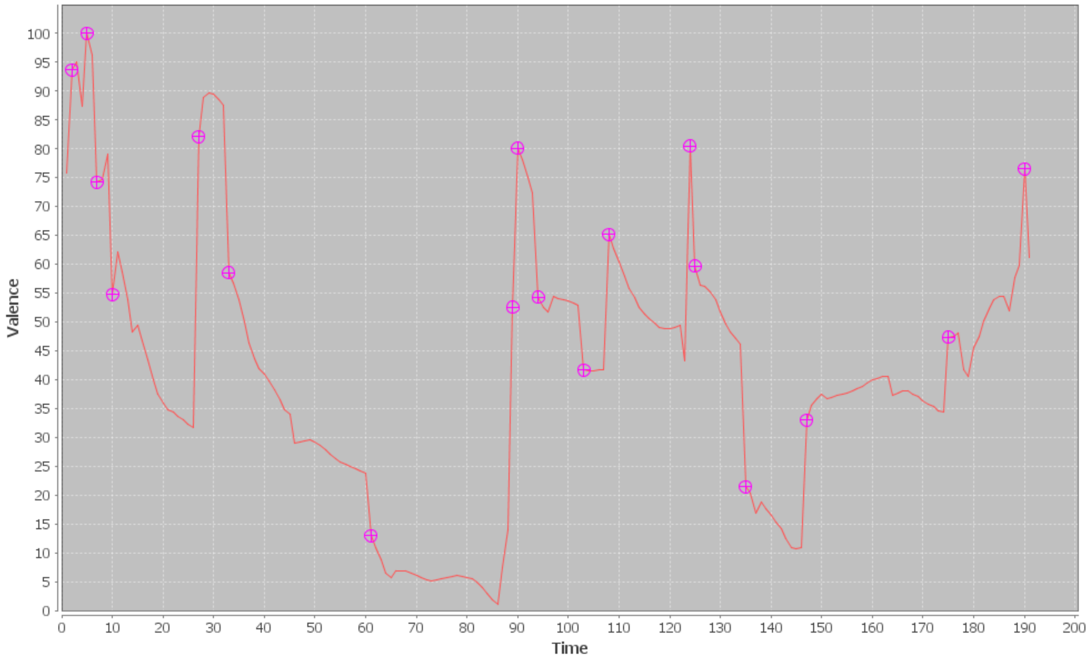


Figure 6.4: Plot of change points extracted using threshold based approach. The threshold used is $Th = 20$

change detection independent of emotion recognition and will later propose to combine them for improving the emotion prediction accuracies. Given the features $\boldsymbol{X_t}$ at time t and the corresponding label Y_t, change detection aims to find a $\hat{t}$ such that $p(Y_t|\boldsymbol{X_t}, t < \hat{t}) \neq p(Y_t|\boldsymbol{X_t}, t > \hat{t})$. But empirically annotating a stream of data with change points for ground truth is a difficult task and there is no publicly available emotion database that contains annotated change points. In this work we define two ways to describe a change point, a) Threshold based and b) CuSum based which are described below.

The AVEC 2012 dataset used for emotion recognition is used for change detection as well. While predicting changes in emotions a value 1 indicates that a major change has occurred and 0 indicates that there has not been a significant change in the emotion. Since the AVEC 2012 datasets is not annotated with changes we used a

Threshold based methodology to annotate change. In this methodology, an n^{th} emotion time series $\boldsymbol{Y_n}$ (i.e. a video annotated with emotions at each frame) is considered and the let min_n and max_n be the minimum and maximum values of emotion in this video. The entire range of values within [min_n,max_n] is scaled to a uniform range of [1,100] (please note that any range can be considered but we chose this one for consistency throughout this work). The new set of emotions are considered and the differences between consecutive values are calculated as $Y_{nt}^{diff} = Y_{nt} - Y_{nt-1}$. For a given threshold $Th \in$ [1,100] all the points $\hat{t} \ni Y_{n\hat{t}}^{diff} > Th$ are considered as change points. Figure 6.4 shows a sample of change points extracted for valence dimension using a threshold $Th = 20$. The drawback of using this methodology is the selection of a threshold which has to chosen empirically.

Each document in an audio sequence or a video is supplied with the intensity of emotions in 4 dimensions. The differences in the intensities for each document in comparison with its predecessor have been extracted. All these differences (from training data) have been scaled to a range of [1,100] and a threshold is applied to all documents. Documents whose difference exceeds the threshold is assigned 1 and 0 otherwise. Due to unavailability of ground truth, Area Under (ROC) Curve has been used as a test metric. Due to imbalance in class sizes F1-Score is also used as a comparative metric.

To train the topic models α= 0.1, β=0.02, γ=0.1 (a parameter for TLTM model), K= 30 have been used. These parameters can be varied but for a comparative analysis of topic models these parameters have been chosen empirically. The threshold that is used to indicate if a change has occurred or not, is also a parameter. The results using different thresholds ranging from 10 (too many changes) to 50 (few changes) are produced. Since LDA and TLTM are unsupervised, they are independent of the dimensions and thresholds. Therefore we trained a single LDA and TLTM model each for all dimensions and thresholds. But since ATTM model takes the change in the dimension intensity into the learning process and one model is for each dimension and for each threshold has been learnt. {10,20,30,40,50} have been used as possible thresholds and thus have trained 20 models to cover all dimensions and thresholds. For each threshold the topic features are extracted from 31 training sequences using three topic models followed by two regression algorithms viz Linear and Support Vector regression (with Radial Basis Function Kernel) that are trained using the topic features. These training models are then used to extract the topic features form 32 test sequences followed by affect dimension prediction by regression models. For each combination of three topic models and two regression algorithms the TP (true positive), FN (false negative), FP (false positive) and TN (true negative) on 32 test sequences are extracted.

Audio Based Results

The sample ROC curves of the three topic models and Linear regression for a threshold value of 30.0 are plotted in Figure 6.3. It can be observed that for this particular threshold the proposed ATTM model has better area under curve for all the dimensions. As shown in Table 6.1 (a), the AUC values of ATTM are greater than the other two models irrespective of the regression algorithm and over all dimensions. Also Linear regression has performed better than the SVR method for audio features. In comparison to video, audio features performed better for the Power and Valence dimensions (highlighted in red). We have also tested Linear regression by changing the ridge parameter from 0 through 1000 but any noticeable change was not observed in the performance. Since the number of changes (or 1's) are very skewed compared to the 0's the F1 measures have been plotted along with the accuracies of predicting changes for a threshold 20, in the left plot of Figure 6.5 (b). It can be seen that

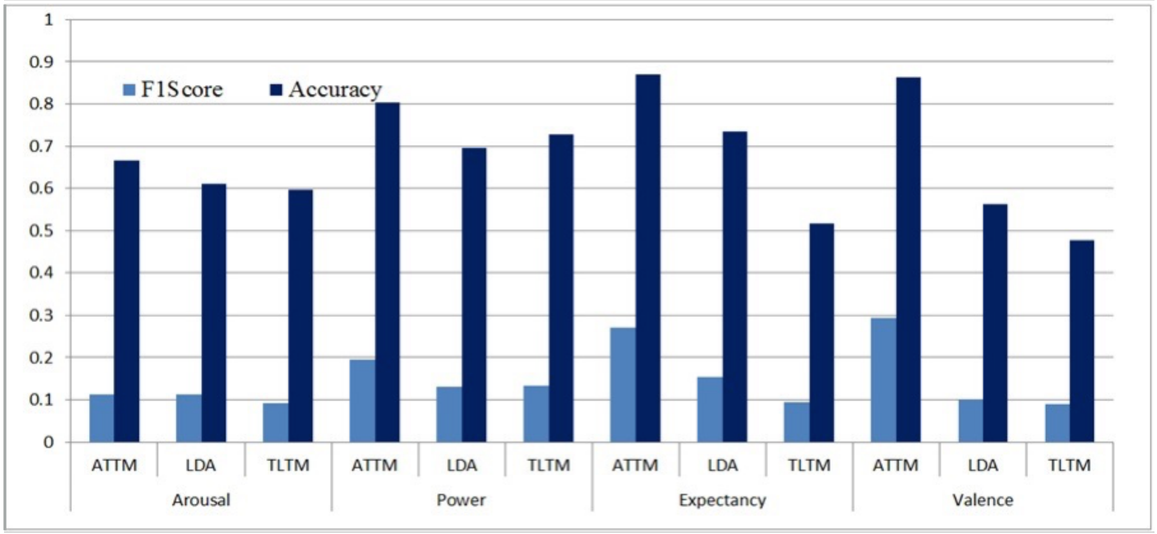


Figure 6.5: F1- Measures and Accuracies from AVEC data using Audio features with Linear Regression, across all thresholds

the ATTM model performs better than other topic models in terms of F1 scores and accuracies as well.

Video Based Results

The video based topic features were extracted using the same set of topic model parameters that have been used in modeling audio. The Areas Under Curve have been calculated for all thresholds and the mean AUCs of video data over all dimensions and thresholds is shown in Table 6.1. In contrast to audio features, for video, SVR performed better than linear regression. This indicates that audio features have a

Table 6.1: Mean AUCs obtained from ATTM , LDA and TLTM models on Audio and Video data with Linear and SVM Regression (SVR). Mean and Variance are calculated using AUCs across 5 thresholds.

Dimension	TopicModel	Mean AUC (Variance)			
		Linear		SVR	
		Audio	Video	Audio	Video
Arousal	ATTM	70 (0.2)	50 (0.1)	68 (0.1)	**85** (0.3)
	LDA	65 (0.1)	49 (0.0)	65 (0.1)	63 (2.4)
	TLTM	61 (0.1)	47 (0.1)	61 (0.1)	75 (2.5)
Expectancy	ATTM	77 (0.3)	50 (0.1)	76 (0.1)	**87** (0.9)
	LDA	72 (0.1)	49 (0.0)	72 (0.2)	**87** (0.9)
	TLTM	62 (0.0)	51 (0.0)	61 (0.1)	**87** (0.8)
Power	ATTM	**79** (0.2)	48 (0.1)	76 (0.1)	72 (0.0)
	LDA	75 (0.1)	46 (0.1)	72 (0.0)	72 (0.0)
	TLTM	74 (0.2)	46 (0.0)	70 (0.2)	72 (0.0)
Valence	ATTM	**80** (0.3)	49 (0.0)	63 (0.2)	60 (0.9)
	LDA	60 (0.0)	48 (0.0)	62(0.0)	49 (0.1)
	TLTM	57 (0.1)	48 (0.0)	57 (0.0)	50 (0.1)

linear dependency on the emotions whereas video features have a non linear dependency. In comparison to audio, video features performed better for the Arousal and Expectancy dimensions (highlighted in red). It is interesting to observe that all the topic models performed equally well for expectancy and power dimensions. One of the reasons for this is that the frequency of changes in both power and expectancy are very less compared to arousal and valence. Due to small number of change predictions to be made, the topic models have performed equally good. The F1 scores

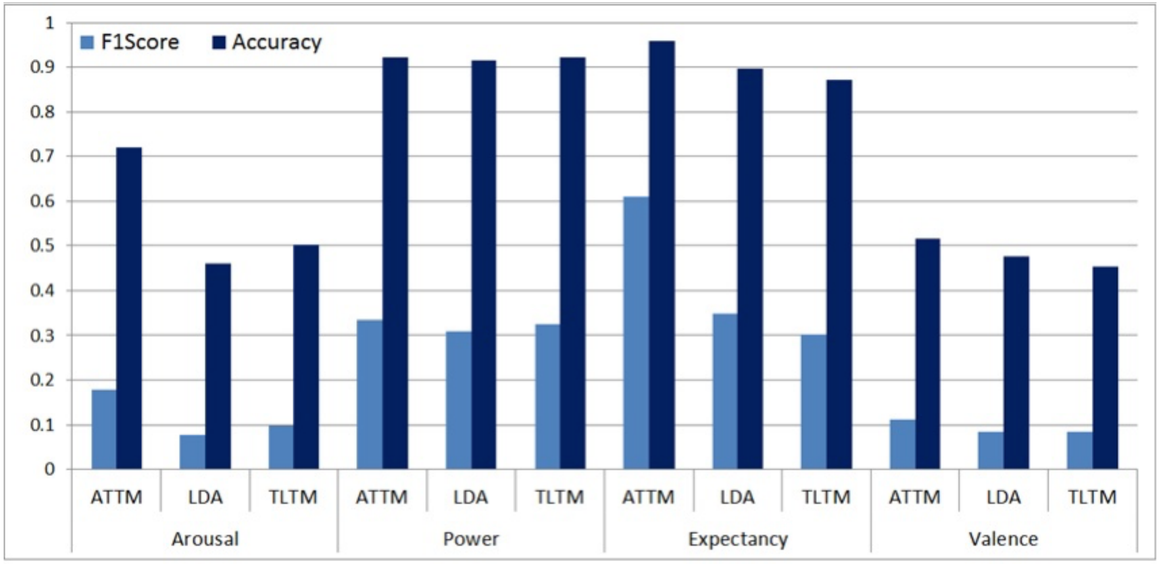
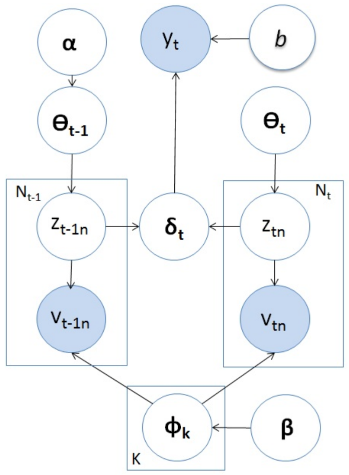


Figure 6.6: F1- Measures and Accuracies from AVEC data using Video features with SVM Regression, across all thresholds

and accuracies of video features with SVR for threshold 20, are plotted in right plot of Figure 6.6 (b). The ATTM model has better accuracies and F1-scores over the other topic models. Although all the topic models gave the same AUC values for expectancy and power, in Figure 6.6(b) we see that ATTM has higher F1 scores and accuracies.

6.2 Supervised LDA for Change Detection (SLDACD) model

ATTM model is a complex model with many variables and the concept of dummy topic assignment may have a negative effect past few iterations. If the sampling is



t : Document at current time t
N_t : # of words in document t
K : # of Topics
$\boldsymbol{z}_t$: N_t-vector of topic assignments to all words in t
$\boldsymbol{v}_t$: N_t-vector of all words that belong to t
$\boldsymbol{\varphi}_k$: Multinomial distribution of terms per topic k
$\boldsymbol{\Theta}_t$: Multinomial distribution of topics per document t
$\boldsymbol{\beta}$: Dirichlet prior vector of $\boldsymbol{\varphi}_k$
$\boldsymbol{\alpha}$: Dirichlet prior vector $\boldsymbol{\Theta}_t$
$\boldsymbol{\delta}_t$: K-vector of real values indicating similarity of K topics in documents *t-1* and *t*
y_t : 0 or 1 indicating if change occurred at time t
$\boldsymbol{b}$: Regression Coefficients to predict y_t using $\boldsymbol{\delta}_t$

Figure 6.7: Graphical model for SLDACD model with explanation of notation used

continued beyond certain number of iterations, there is a chance that all words are assigned to the single dummy topic. To avoid this and to reduce the complexity we proposed another topic model supervised LDA for change detection (SLDACD) which is based on the supervised LDA discussed in Chapter 5.

Figure 6.7 shows the graphical model for SLDACD model with the notation. In this model we assume that at a given time step, the change in emotion depends on both current and previous normalized topic probabilities $\bar{z}_{t-1}$ and $\bar{z}_t$ through the variable $\boldsymbol{\delta_t}$. y_t is the observed variable assuming 0 or 1 depending on whether change has occurred, and is modeled as a Logistic function of $\boldsymbol{\delta_t}$ and $\boldsymbol{b}$ where $\boldsymbol{b}$ are the regression coefficients. For each topic k, δ_{tk} is calculated from z_{t-1k} and z_{tk} as: (i) Scale z_{t-1k} and z_{tk} from $[0, 1]$ to $[min, max]$ where $min, max > 1$; (ii) Calculate the differences $a_1 = |z_{t-1k} - z_{tk}|$ and $a_2 = max - a_1$; and (iii) Calculate $\delta_{tk} = a_2/(a_1 + a_2)$. Note that for each topic k, δ_{tk} is the expected value of the $Beta(a_1, a_2)$ distribution. If the topic probabilities z_{t-1k} and z_{tk} are very close then a_1 increases and a_2 decreases and thus the distribution $Beta(a_1, a_2)$ will be right-skewed with a mean that moves towards 1 and vice versa. Thus δ_{tk} can be seen as the probability of z_{t-1k} and z_{tk}

being similar. This topic wise similarity probability vector is used as feature vector to predict change at current time step.

Similar to the one in SLDA, the conditional distribution of $\boldsymbol{z}$ over the rest of the variables is given by:

$$p(\boldsymbol{z}|\boldsymbol{\alpha}, \boldsymbol{\beta}, \sigma^2, \boldsymbol{v}, \boldsymbol{y}, \boldsymbol{b}) \quad \propto \quad p(\boldsymbol{v}|\boldsymbol{z}, \boldsymbol{\beta})p(\boldsymbol{z}|\boldsymbol{\alpha})p(\boldsymbol{y}|\boldsymbol{z}, \boldsymbol{b}, \sigma^2)$$

and the distribution of change indicator variable y_t which is a logistic function over the variables is given by

$$p(y_t = 1|\boldsymbol{\delta_t}, \boldsymbol{b}) \sim \frac{1}{(1 + e^{(} - \boldsymbol{b\delta_t}))}$$

Using the collapsed Gibbs Sampling approach discussed in Section 5.1.1 and [15], the topic z_{tn} of the n^{th} word w of document t is sampled using the following equation:

$$p(z_{tn} = k|\boldsymbol{v}, y_t, \boldsymbol{z}_{\neg tn}, \alpha, \beta) \sim (n_{tk}^{\neg n} + \alpha)\frac{(n_{vk}^w + \beta)}{(n_k + V\beta)}p(y_t|\boldsymbol{\delta_t^{\neg w}}, \boldsymbol{b})$$

where $p(y_t = 1|\boldsymbol{\delta_t^{\neg w}}, \boldsymbol{b}) \sim 1/(1 + exp(\boldsymbol{b}\delta_t^{\neg w})exp(b_k/N_t))$. $n_{tk}^{\neg n}$ is the # of times topic k is assigned to document t excluding current word, n_{vk}^w is the # of times the word w is assigned to topic k and $\boldsymbol{\delta_t^{\neg w}}$ is the $\boldsymbol{\delta_t}$ vector that does not include the current topic assignment to the document. At the end of each Gibbs iteration, all of the $\mathbf{z}$ are sampled and the $\boldsymbol{\delta}$ vectors are calculated using the method described earlier. An EM type of algorithm is used to estimate all the variables $\boldsymbol{z}$, $\boldsymbol{\delta}$ and $\boldsymbol{b}$ as follows:

E-Step: Variables $\boldsymbol{z}$, $\boldsymbol{\delta}$ are estimated

M-Step: Coefficients $\boldsymbol{b}$ are estimated using Logistic regression over $(\boldsymbol{\delta_t}, y_t)$ pairs.

6.2.1 Experiments and Results

AVEC12 dataset has been used for evaluation SLDACD model on emotion change detection. Unlike ATTM model that specifically works for threshold based change

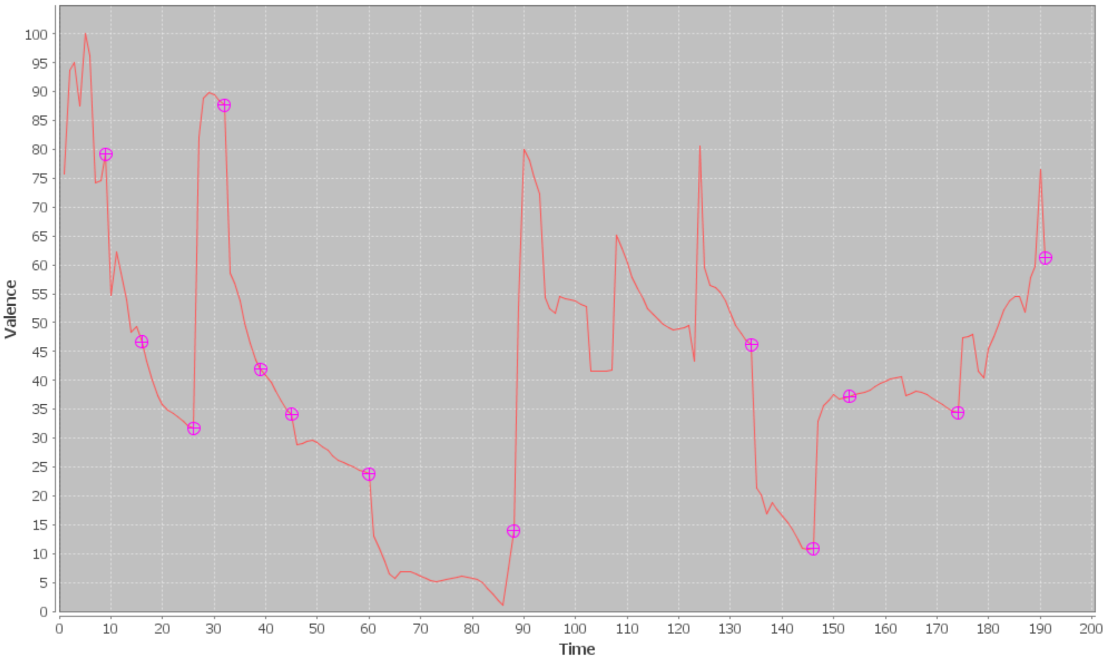


Figure 6.8: Plot of change points extracted using CuSum based approach. The confidence used is $Conf = 99\%$

detection, SLDACD is used to model annotations that consider the entire sequence. We annotated the dataset using a method called CuSum which is detailed below.

CuSum based change annotation:

CuSum stands for Cumulative Summation and this methodology is used to extract change points by cumulatively summing the emotions and was proposed by Wayne [65]. For an n^{th} emotion time series $\boldsymbol{Y_n}$, the cumulative sums $\boldsymbol{S_n}$ are calculated as $S_{nt} = S_{nt-1} + Y_{nt} - \bar{Y_n}$ where $\bar{Y_n}$ is the mean of current video/series and S_{n0}= 0. A statistic $S_{n,diff} = S_{n,max} - S_{n,min}$ is calculated for each video $\boldsymbol{Y_n}$ and its bootstrap samples $\boldsymbol{Y_n^{boot}}$ (permutations of the original sequence $\boldsymbol{Y_n}$). For a given confidence level $Conf$ the statistic $S_{n,diff}$ is compared with the statistics $S^{boot}_{n,diff}$ and confidence is calculated as $100 * \frac{(\#\text{of samples whose statistic } S^{boot}_{n,diff} < S_{n,diff})}{(\#\text{ of bootstrap samples })}$. If this confidence is greater

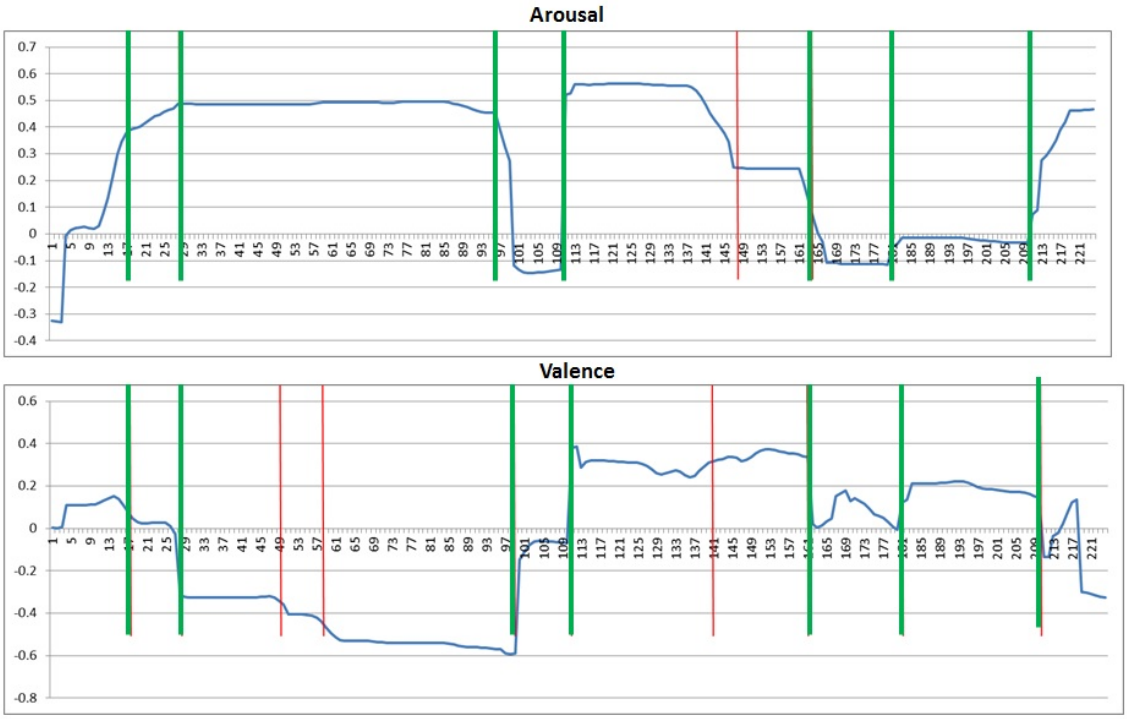


Figure 6.9: Plot of change points extracted using CuSum based approach on arousal and valence for the same video

than the required confidence $Conf$ then a change is assumed to have occurred. And the change point is calculated as $\hat{t} = \max\limits_{t} |S_{nt}|$ i.e. the time step that has the maximum value of statistic. This implies that a change point is one that has the maximum amount of change over the average in comparison to the rest. Once a change point $\hat{t}$ is extracted the series is split to two sub-series and recursively the change points are extracted from each of the sub-series.

Figure 6.8 shows the plot of change points extracted using the CuSum algorithm. There are many difference between the threshold and Cusum based approaches. Firstly Cusum is more unsupervised approach where the selection of a threshold is not needed. Secondly while threshold based method only uses the emotion from previous time step, this approach uses the average information from entire series.

So threshold approach can handle streaming data but Cusum requires batch data. Another interesting observation regarding Cusum method is that for many videos, most of the change points detected for arousal were matching with those detected for valence. This implies that the dimensions are very correlated in terms of changes in the signals. A sample plot for a given video is shown in Figure 6.9 where the green lines indicate the change points that have matched for both arousal and valence.

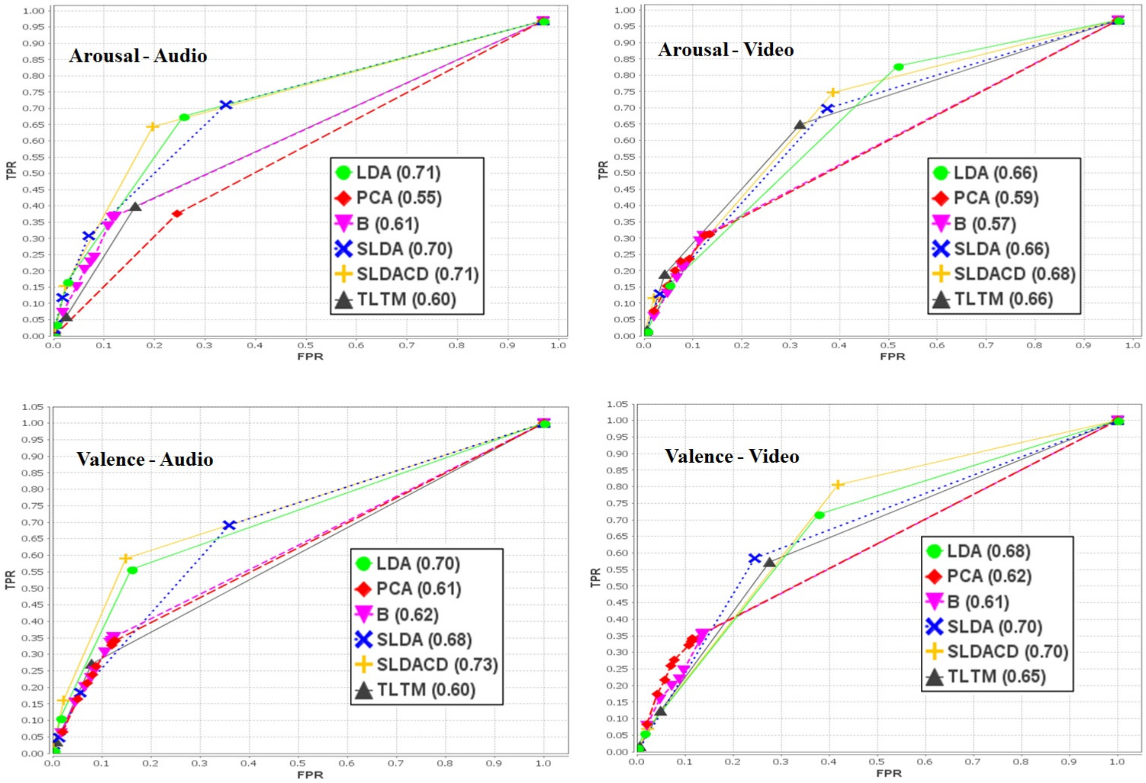


Figure 6.10: ROC curves of topic models in comparison to baseline and PCA features for Arousal and Valence using Audio and Video features

Results

In this work only arousal and valence have been considered as they are considered to be more informative. We have used LDA, sLDA, TLTM [34] (a topic model where the current word is influenced by both $\boldsymbol{\theta_t}$ and $\boldsymbol{\theta_{t-1}}$), the proposed SLDACd, PCA and Baseline (B) methods. SVM Regression with Radial basis function (SVRR) kernel has been used with the baseline and PCA features to predict the emotions. SLDACd predicts 0/1 directly and for other topic models the topic features and $\boldsymbol{\delta_t}$ are calculated for each document and then used logistic regression for predictions. For Baseline and PCA methods, we used the actual predictions of arousal and valence and extracted changes at different confidence intervals.

The ROC curves are shown in Figure 6.10 with the AUC values in parentheses next to the corresponding model. It can be observed that the Baseline and PCA methods do not perform as well as topic models on change detection, indicating that they possibly do not predict variations in emotions. This is in line with the results and discussion given in Table 4.11 which indicate that the predictions by PCA are so small that the changes are not considered to be significant. The proposed SLDACD method gave the best AUC values for most dimensions except for arousal-audio and valence-video where it performed equally as good as LDA and SLDA. The inclusion of temporal information helped SLDACD to outperform LDA and SLDA and the inclusion of changes into the model in a semi-supervised fashion helped in performing better than TLTM model.

6.3 Conclusions

In this chapter, two new topic models have been proposed, the inference equations have been derived and they have been evaluated for emotion change detection. Two models, ATTM and SLDACD have been used to model local and global changes in the response variables. We have compared their performance against non temporal topic models and other dimension reduction techniques and have shown that these models outperform in their evaluation over change detection. The models and evaluation presented in this chapter have been published at IEEE ICME [35] and NIPS Workshop on Topic Models [36].

Chapter 7

OTHER CONTRIBUTIONS

In Chapters 4 through 6 we have presented and evaluated graphical models over emotion recognition and change detection applications. While we have worked on the theoretical aspects of the pattern recognition models, we also contributed to the Social Interaction Assistant project for people who are blind. This project has been a motivating factor that enabled us to work on emotion recognition and also look into the structure of topic models. In Section 7.1, we will discuss implementation details of a real time emotion recognition system we have built. Another outcome of this research is Java API specifically developed for using Probabilistic topic models for video analysis and we will discuss this ins Section 7.2.

7.1 Social Interaction Assistant

Social Interaction Assistant is a flagship project undertaken at Cognitive Ubiquitous Computing Center (CUbiC) [1] to assist people who are visually impaired in their daily social interactions. Most of the non verbal cues, gestures, emotions and facial expressions that are taken for granted during our conversation are in fact extremely important in guiding our social and personal well being. People who are blind do not have access to this information and there by miss out on many social cues that occur during our conversations, meetings and in public life. Sreekar et al. [32] have done a requirement analysis where blind users have indicated the importance of technologies that they want to have to improve their social well being, Taking cues from these findings we learnt that one of the top priority of blind users *was to know the facial expressions and emotions of other people.* Thus we went ahead to build an Assistive

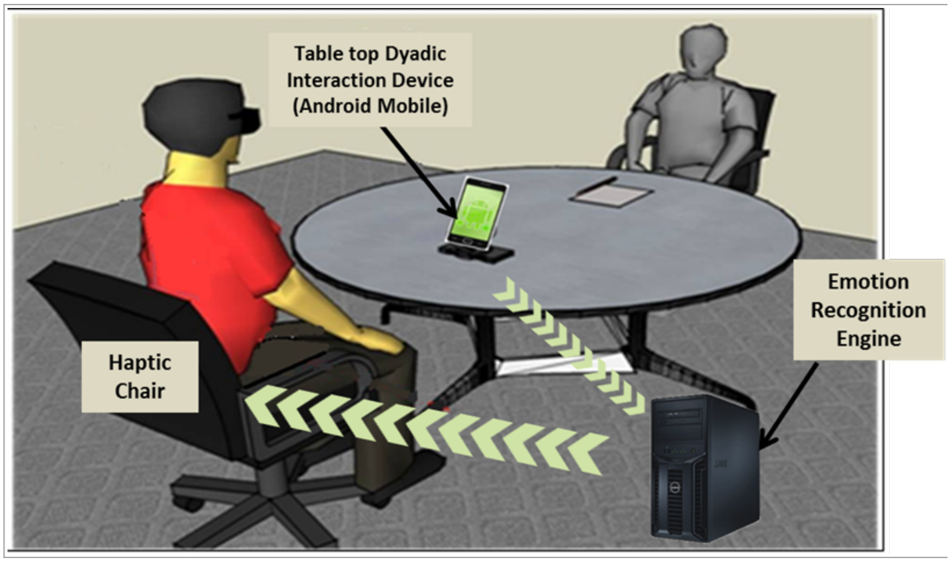


Figure 7.1: System Architecture of the Social Interaction Assistant Project

technology which could help blind users with understanding and perceiving other's emotion states.

7.1.1 System Architecture

We have designed the system architecture for the Social Interaction Assistant (SIA) project as shown in Figure 7.1. The main software of this project is integrating into a Mobile Application that can run on the blind user's smart phone. This device can be placed on the table top set up and the mobile device will track the faces of people with whom the blind user may be having a conversation with or may be interested in. The faces that are tracked are communicated to an Emotion recognition engine that can run on any system with sufficient computing power. It should be noted that in future we will be including the engine in the mobile device itself but this may need the device to have much more computing and battery power. The recognition engine extracts faces from the images and executes fast and reliable emotion recognition algorithms and detects the facial expressions. It then communicates the emotions to the third key component of the application which is Haptic based chair that will vibrate certain patterns that can appropriately deliver the emotion to the user.

We have used Android Development Toolkit to develop the Mobile Application. There are three primary modules within the application, 1) Face Tracker, 2) Device vibrator, and 3) Image Communicator. The Face Tracker tracks and detects faces and we have used OpenCV API to achieve this. In order to help the blind user to fixate on a particular person, we vibrate the device in certain patterns to indicate the facial position with respect to the camera. Finally, the device captures the facial images and communicates to the engine through bluetooth.

Emotion Recognition Engine

Figure 7.2 describes various components of the Emotion Recognition Engine where the the input comes from the mobile device and the output goes to the Haptic chair. The engine continuously polls for images and extracts two different features, 1) geometric features (based on facial landmarks), and 2) appearance features (LBPTop). Once these features are extracted they are passed on to the emotion classification models that can clarify the current image. In this application we have used Naive Bayes and Support Vector Machines as the classifiers for geometric and appearance features. We have used the Intraface API [66] to extract landmarks and LBPTop API [41] to extract LBPTop features. Since it is not meaningful to used the landmarks directly, we have devised a way to extract features from the landmarks.

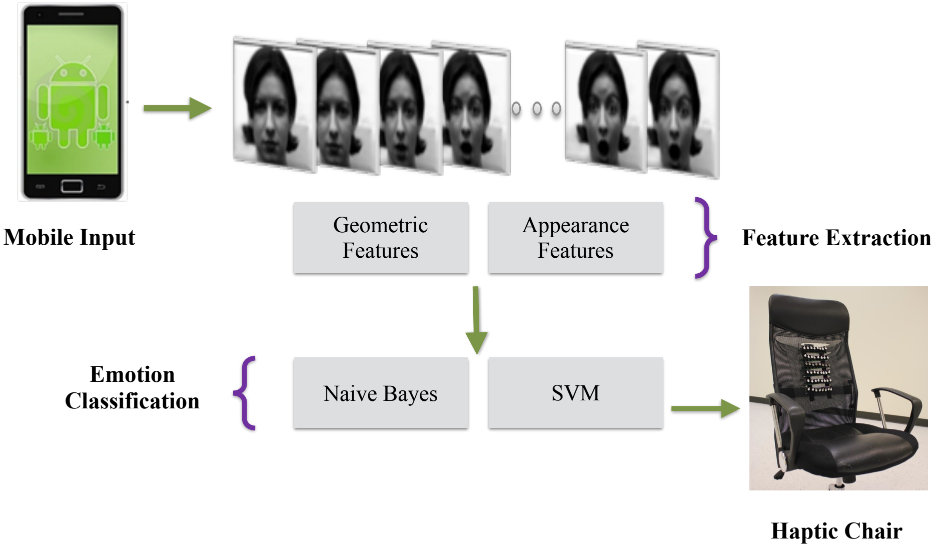


Figure 7.2: Various Components Within the Emotion Recognition Engine

We have extracted a set of 36 (18x2) measurements from the landmark features which are displayed in Figure 7.3. We calculated the distances and the angles between the 18 points shown in the picture. From every frame we extract 36 features and then we find the ratios of the changes in 36 measurements in comparison to a neutral frame. Since it is difficult to obtain a neutral frame for every person we picked up a single neutral frame as reference and trained all our models on those features. This helped us work around the problem and to avoid the scaling, we normalized and scaled every image to the size and scale of neutral image. For LBPTop features, we need a sequence of images and thus we used the preceding 7 frames and extracted the voxel features of a total size 2832 which are obtained from 16 blocks, 3 dimensions and 59 LBP patterns. We used the Cohn Kanade Plus [40] dataset to train classifiers on these features with 7 discrete emotions, anger, happy, disgust, surprise, sad, contempt

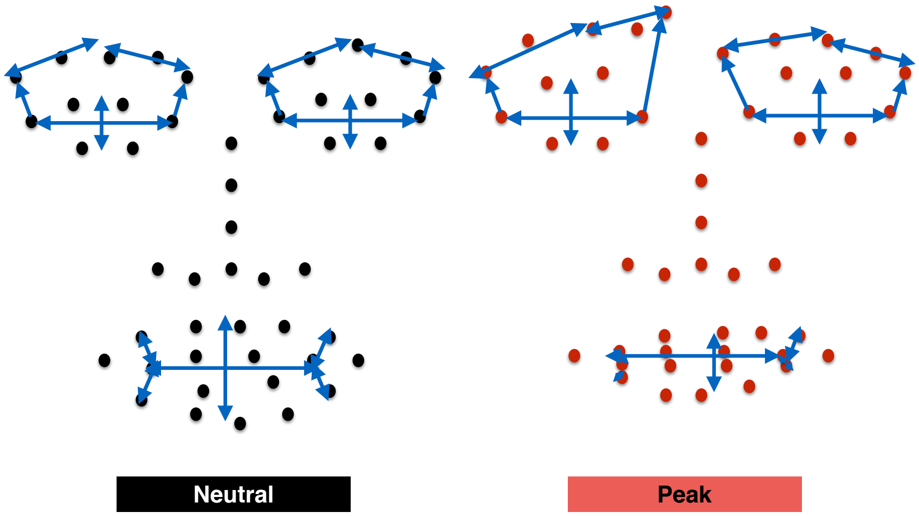


Figure 7.3: Different Features Extracted from Facial Landmarks

and fear. We have used Naive Bayes classifier for landmark features and SVM for LBPTop features.

We performed a 3 fold cross validation for each classifier and we achieved 89% and 100% classification accuracies using Landmark and LBPTop features respectively. Since LBPTop features performed better we have included them in the engine. We have also retained landmarks based classifier so that in cases where we may need both classifiers to enhance the performance. McDaniel et al [43] have worked extensively on haptic patterns that can best convey human emotions and the emotions detected by the engine are communicated to the chair using the patterns explained in [43]. We have a working prototype for the Social Interaction assistant device and we will be performing user studies so that we can get feedback from the blind users and can improve upon the existing features.

7.2 Java API for Topic models for Video Analysis

In this thesis, we have proposed and developed several graphical models to perform video analysis. A major roadblock in implementing or using existing code is the need for consistency across models and inference techniques. There are many open source Latent Dirichlet Allocation based Packages like StanfordNLP (Scala) , Mallet Toolbox (Java), LDABlei (C++) and CRanTopicModels (R) that are available. But since each of them had only particular implementation of certain model in different programming language, it is very difficult to compare and replicate results. And none of these have been written specifically to address video datasets and features. This has motivated us to implement all the topic models from the scratch and it was very useful while benchmarking our results.

7.2.1 Package Structure

We have developed an open source Java based API that contains Map-Reduce based implementations of different topic models . Since our research is about supervised learning, we have also developed wrappers for different regression and classification models based on the open source machine learning package Weka [7]. The major packages and the classes are shown in Figure 7.4. The Dimension Reduction package contain implementations for all the topics models we have presented in this work, LDA, SLDA, RSLDA and DGMM models. We have also implemented PCA and GMM models as one of the dimension reduction techniques. Since each of these are probabilistic models with different inference techniques, we have implementations for Gibbs based sampling and Variational EM algorithms for LDA, SLDA and GMM. This will help researchers to benchmark different algorithms at a go.

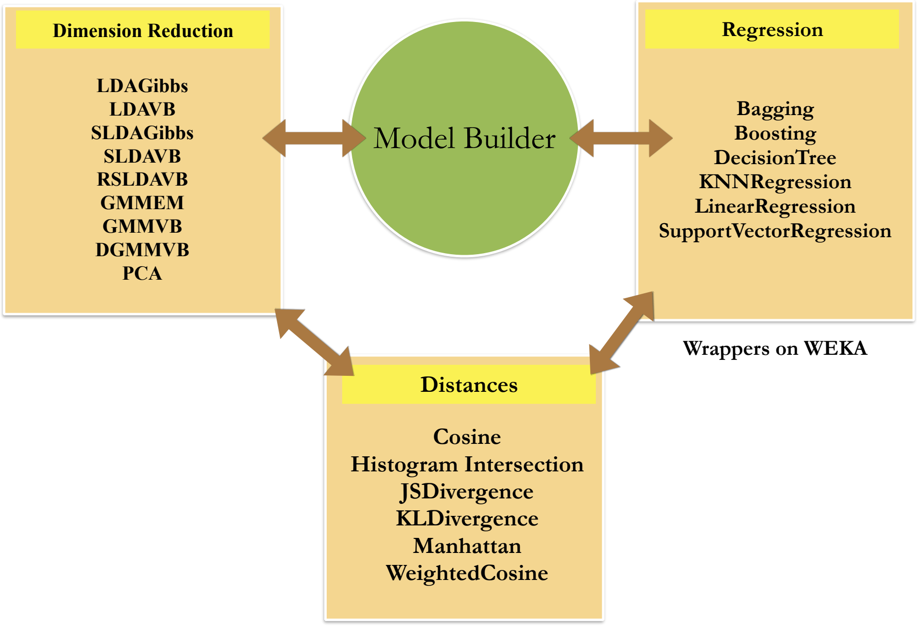
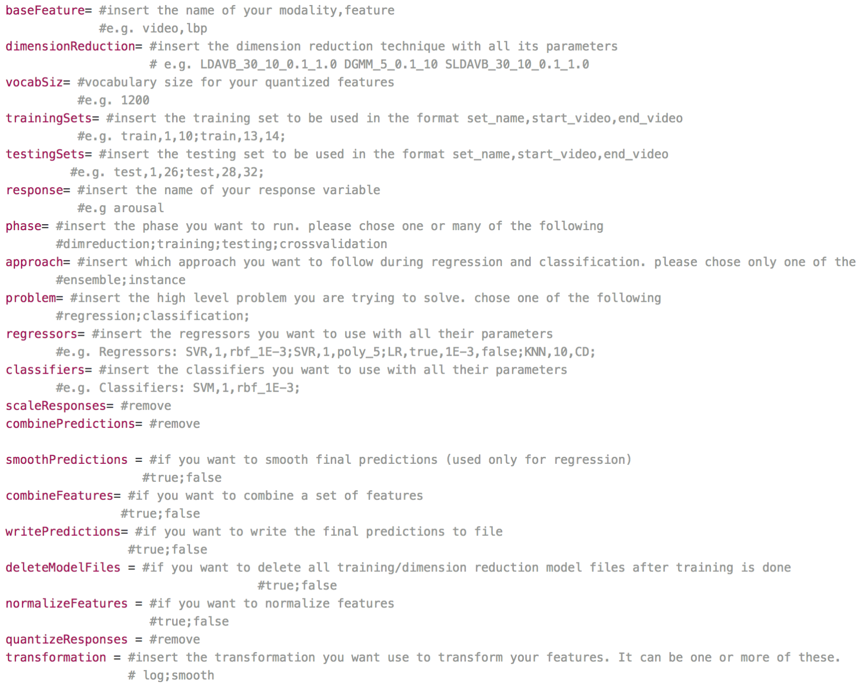


Figure 7.4: Package Structure in Topic Models Java API.

The API also contains another package for regression models like Support Vector Regression, Bagged methods, Linear Regression etc. It should be noted that these are just wrappers for Weka API and we have not implemented the models in general. Since many of these methods may need different Kernels and distances we have implemented Histogram Intersection, Cosine, KL Divergence distance based kernels. To reduce the hassle of providing parameters for each algorithm in a different format, we have created a Parameters.properties file where users can fill in the needed features, parameters and model names and the required models are then automatically loaded. For example to run the LDA model on LBPTop features with 10 topics, $\alpha = 0.1$ and 10 iterations we fill in the properties file as:

```
baseFeature = video, lbptop

dimensionReduction = LDAVB_10_10_0.1
```



```
baseFeature= #insert the name of your modality,feature
             #e.g. video,lbp
dimensionReduction= #insert the dimension reduction technique with all its parameters
                    # e.g. LDAVB_30_10_0.1_1.0 DGMM_5_0.1_10 SLDAVB_30_10_0.1_1.0
vocabSiz= #vocabulary size for your quantized features
          #e.g. 1200
trainingSets= #insert the training set to be used in the format set_name,start_video,end_video
          #e.g. train,1,10;train,13,14;
testingSets= #insert the testing set to be used in the format set_name,start_video,end_video
         #e.g. test,1,26;test,28,32;
response= #insert the name of your response variable
          #e.g arousal
phase= #insert the phase you want to run. please chose one or many of the following
       #dimreduction;training;testing;crossvalidation
approach= #insert which approach you want to follow during regression and classification. please chose only one of the
       #ensemble;instance
problem= #insert the high level problem you are trying to solve. chose one of the following
       #regression;classification;
regressors= #insert the regressors you want to use with all their parameters
       #e.g. Regressors: SVR,1,rbf_1E-3;SVR,1,poly_5;LR,true,1E-3,false;KNN,10,CD;
classifiers= #insert the classifiers you want to use with all their parameters
       #e.g. Classifiers: SVM,1,rbf_1E-3;
scaleResponses= #remove
combinePredictions= #remove

smoothPredictions = #if you want to smooth final predictions (used only for regression)
                   #true;false
combineFeatures= #if you want to combine a set of features
                #true;false
writePredictions= #if you want to write the final predictions to file
                 #true;false
deleteModelFiles = #if you want to delete all training/dimension reduction model files after training is done
                                   #true;false
normalizeFeatures = #if you want to normalize features
                    #true;false
quantizeResponses = #remove
transformation = #insert the transformation you want use to transform your features. It can be one or more of these.
                 # log;smooth
```

Figure 7.5: Parameters properties file that can be used to specify any features, options and models

```
phase = dimreduction
```

Since we have implemented the EM algorithms using a map-reduce framework, users will benefit from multi core or multi node systems or super computing machines where documents are shared between the cores. It has to be noted that some of the algorithms like Gibbs sampling where the sampling a document depends on the rest of corpora, we will not be able to take advantage of parallel systems. The API allows for threes phases, viz. *training*, *texting* and *dimension reduction* but the user needs to ensure that the required phases have been executed in the required sequence. In Chapter 8 we will summarize all the work done as part of this thesis and propose some future work in this area of research.

Chapter 8

CONCLUSIONS

With increasing availability of multi modal data over the web and through publicly collected datasets, there is a dire need to analyze the corpus using reliable and faster techniques. Like text and web data, video and audio features are very high dimensional and contain information that may not be relevant to the task at hand. These tasks could be activity recognition, emotion recognition, video retrieval or clustering. In this thesis we have focused our attention on Continuous emotion recognition where the predicted values are continuous emotion dimensions, arousal and valence, and the predictions are made at every frame of a video.

Dimension reduction plays a very important role in understanding features and training regression models. In this thesis we have touched upon a new framework for dimension reduction based on probabilistic topic models. Latent Dirichlet Allocation is a dimension reduction technique proposed for text datasets to achieve efficient clustering of documents. We have applied these models to the context of dimension reduction of multimodal video data. In Chapter 4 we discussed the graphical structure of topic models and the learning and inference methodologies estimate parameters. We discussed the Variational Expectation maximization technique in detail and have illustrated a Map-Reduce framework to implement it. We have then shown that the new *topic* features extracted from various video and audio features can be interpreted and visualized. In Section **??** we have provided evidence to show that LDA outperforms traditional PCA technique in the context of continuous emotion recognition. Along with PCA which is a linear projection model, we have also compared LDA to

KernelPCA which is non linear version of PCA. LDA has shown better performance in comparison to KernelPCA models also.

Since LDA models cannot handle continuous features which occur in most of video and audio data, we have proposed a new Dirichlet Gaussian Mixture Model (DGMM) that can be used without quantizing features. DGMM assumes a continuous probability density for the topic-term space and we derived the Variational EM equations to infer the parameters. We have shown that DGMM which falls under the category of techniques like PCA that work on continuous feature space, has again outperformed PCA model. And we have also see that DGMM gave comparable performance to LDA for emotion recognition though the model training is extremely slow when compared to LDA model. Dimension reduction techniques project data to a space where the the variance between the features is maximized but they don consider the label or response space. There has not been much work done in the area of supervised dimension reduction for multi modal data where the response variable is continuous. In Chapter 5 we have given details about Supervised LDA model (SLDA) which is a supervised extension to LDA model.

SLDA models the continuous response variable within its graphical structure as a linear combination over the empirical topic distributions. We have derived the Variational EM inference for SLDA and illustrated the likelihood changes and the effect of EM algorithm on the training error. We evaluated SLDA models on the Avec12 dataset and have shown that they consistently do better than LDA across all features , regressors and dimensions. Since SLDA model tends to overfit topics to emotions, we have incorporated regularization into SLDA's graphical structure called RSLDA (Regularized SLDA). In Section 5.2 we derived the EM equations for the RSLDA model where we included the hyper prior parameters in inference. We have illustrated how the inherent structure of RSLDA ensures a better generalization than

SLDA model. We have evaluated arousal and valence prediction for RSLDA model in Section 5.2.1.

Since RSLDA is a supervised model we compared its performance with two supervised reduction techniques, Correlation Based Supervised Feature selection (CSF) and Supervised PCA (SPCA) models. In Section 5.2.1 we have validated that RSLDA performs very well in comparison to both CSF and SPCA models. Throughout our studies we have found an inherent pattern that video based topics tend to perform well for valence prediction and audio topics for arousal. To make use of this pattern we have multimodal fusion of topic features from audio and video modalities. The results are extremely encouraging and we were able to achieve good performance on both arousal and valence prediction.

Other outcomes of this work have been creating a Social Interaction Assistant application for people who are visually blind where we convey the behavior state of people to visually impaired through emotion recognition models and haptic interface devices. We have built a Mobile Application that can decipher the emotions and send the information to blind users. Another major outcome of this work is a Java API for Topic models specifically tuned to video data. We have made it available to public through GitHub [4] which can be used to train different topic models discussed in this thesis for various applications at hand.

In Chapter 6 we have introduced two new topic models Adaptive Temporal Topic model (ATTM) and Supervised LDA for Change Detection (SLDACD) model. In Chapters 4 and 5 we have evaluated models for emotion recognition but emotion change detection is another promising application. In Chapter 6 we have addressed emotion change detection using topic models where we have incorporated the temporal component into the model. We have modeled pairs of documents instead of considering them independent and used two different generative models to learn the relationship between topic changes and emotion changes. ATTM and SLDACD, each have performed better than the non temporal topic models and PCA in detecting changes to emotions.

Along with these theoretical contributions, in Chapter 7, we have discussed two other outcomes of this thesis. We have developed a Social Interaction Assistant device that assists people who are blind. At the core of this device lies the emotion recognition engine where we have trained classification models on 7 discrete emotions using CKPlus database [40]. Another outcome of this research is Java API that contains topic model implementations. All the models that have been evaluated have been made available to research community along with API for supervised learning specific to video and audio data.

8.1 Future Work

There is a lot of learning and a lot of scope in the application of topic models to video analysis. Its graphical structure is very useful that we can incorporate the modalities and components that are needed by the application at hand. Some of the models that we have not explored in this thesis are topic models that can consider the spatial relationships between features and supervised continuous topic models. There is lot of promise in application of topic models as dimension reduction techniques for video data and this thesis evaluated this promise and proved it to be correct.

In Chapter 4 we have discussed continuous unsupervised models but were not able to extend it to a supervised setting due to the time it takes to extract topic features. In future we would like to improve this model by 1) modeling groups of continuous features instead of single features, 2) providing an online learning mechanism that can update parameters by feeding in one document after other, and 3) developing a supervised continuous model that has the advantages of both RSLDA and DGMM models.

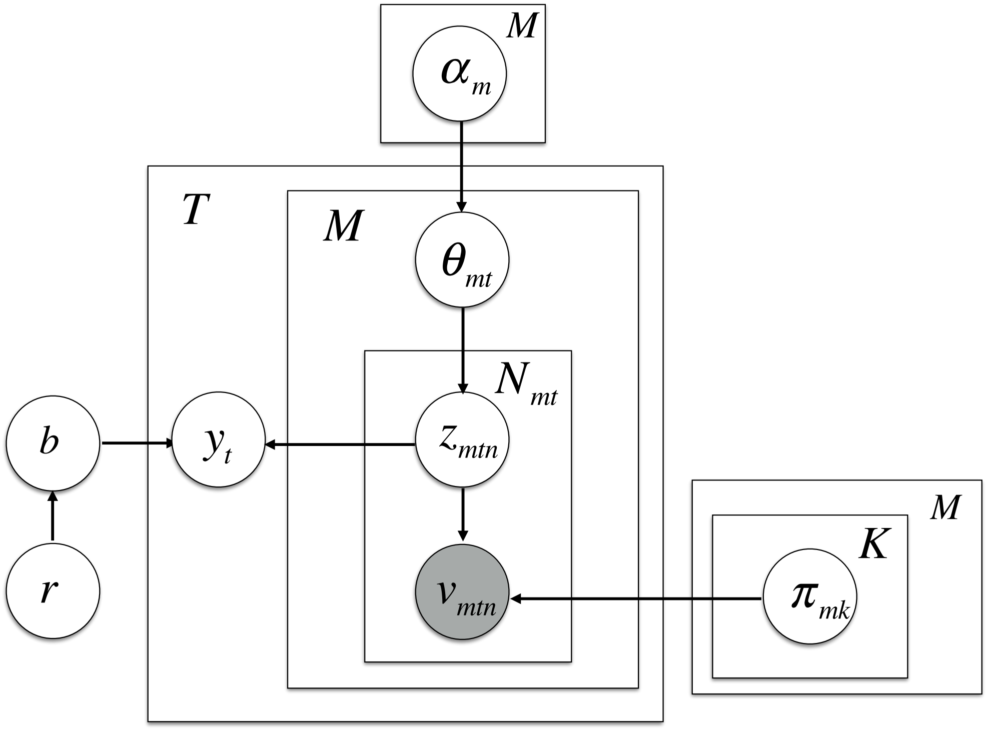


Figure 8.1: Graphical Structue of Multi Modal Supervised LDA Model.

Regularized Multimodal SLDA model:

At the end of Section 5.2.1 we have evaluated multimodal fusion of features and their good performance has motivated us to think of a new multi modal topic model. Researchers have worked on multi modal topic models earlier [48] but in the context of image annotation where one modality is the image and the other modality contains text labels. And the generative model assumes that the actual topics are related to the labels and the labels generate image features. But in the context of multimodal video analysis we need to model audio and video models jointly as shown in Figure 8.1.

By analyzing the multimodal fusion results from Section 5.2.1, it can be noted that LR regressor has performed the best and for RSLDA when we say LR it does not mean a separate regression but the linear regressor within the RSLDA model. This indicates that the linear predictors within RSLDA model perform better when topics from different modalities are combined. This gives an interesting idea about a probable extension of RSLDA model to a Multimodal model which we call the Regularized Multimodal SLDA (RMMSLDA). Even though we have not tested this model we would like to close this section by introducing the model's structure. Multimodal Topic model shown in Fgiue 8.1 extends the structure of RSLDA model but we find a new plate notation and a new variable M. M is the number of feature modalities being considered, e.g. if LM, LBPTop and Audio, all three are considered then $M = 3$. Thus every document is indexed by the feature id m and then the document id n. The Variational EM algorithm will be similar to RSLDA model but the major change would be that while calculating the linear predictors $\boldsymbol{b}$ we will consider the empirical topic features from all modalities. We have not implemented this model but would like to explore these models in future.

In this thesis we have restricted our analysis to emotion datasets but we would like to extend our evaluation to other datasets like activity recognition, image retrieval and video clustering in future.

REFERENCES

[1] Center for cognitive ubiquitous computing. http://cubic.asu.edu/. Online; accessed February-2015.

[2] Dirichlet distribution. http://en.wikipedia.org/wiki/Dirichlet_distribution. Online; accessed February-2015.

[3] A european network of excellence in social signal processing. http://sspnet.eu/. Online; accessed February-2015.

[4] Github repository. https://github.com/cubic-asu/topic-models-for-video-analysis. Online; accessed February-2015.

[5] Sift features. http://www.vlfeat.org/matlab/vl_sift.html. Online; accessed February-2015.

[6] Softmax function. http://en.wikipedia.org/wiki/Softmax_function. Online; accessed February-2015.

[7] Weka. http://www.cs.waikato.ac.nz/ml/weka. Online; accessed February-2015.

[8] N. Alugupally, A. Samal, D. Marx, and S. Bhatia. Analysis of landmarks in recognition of face expressions. *Pattern Recognition and Image Analysis*, 21(4), 2011.

[9] E. Bair, T. Hastie, D. Paul, and R. Tibshirani. Prediction by supervised principal components. *Journal of the American Statistical Association*, 101(473), 2006.

[10] E. Barshan, A. Ghodsi, Z. Azimifar, and M. Z. Jahromi. Supervised principal component analysis: Visualization, classification and regression on subspaces and submanifolds. *Pattern Recognition*, 44(7):1357–1371, 2011.

[11] D. P. Bertsekas. *Nonlinear Programming.* Athena Scientific, 1995.

[12] D. M. Blei and J. D. McAuliffe. Supervised topic models. *Arxiv.org*, 2010.

[13] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. M. L. R.*, Mar. 2003.

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

[15] J. Chang. *Uncovering, Understanding and predicting links.* PhD thesis, Princeton University, 2011.

[16] J. Chang and D. M. Blei. Relational topic models for document networks. In *International Conference on Artificial Intelligence and Statistics*, pages 81–88, 2009.

[17] W. Chong, D. Blei, and F.-F. Li. Simultaneous image classification and annotation. In *CVPR 2009*.

[18] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *IEEE Transactions on pattern analysis and machine intelligence*, 23(6):681–685, 2001.

[19] J. Dean and S. Ghemawat. Mapreduce: Simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113, 2008.

[20] S. Deerwester. Improving information retrieval with latent semantic indexing. 1988.

[21] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.

[22] A. Dhall, A. Asthana, R. Goecke, and T. Gedeon. Emotion recognition using phog and lpq features. In *Automatic Face Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, pages 878–883, 2011.

[23] P. Ekman and W. V. Friensen. *The facial action coding system (FACS): A technique for the measurement of facial action.* Consulting Psychologists Press, 1978.

[24] L. Fei-Fei and L.-J. Li. What, where and who? telling the story of an image by activity classification, scene recognition and object categorization. In *Computer Vision*. Springer Berlin / Heidelberg, 2010.

[25] R. A. FISHER. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936.

[26] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6):721–741, 1984.

[27] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1999.

[28] P. Hu, W. Liu, W. Jiang, and Z. Yang. Latent topic model based on gaussian-lda for audio retrieval. In *Pattern Recognition*, volume 321 of *Communications in Computer and Information Science*, pages 556–563. Springer Berlin Heidelberg, 2012.

[29] Y. Hu, A. John, F. Wang, and S. Kambhampati. Et-lda: Joint topic modeling for aligning events and their twitter feedback. In *AAAI*, volume 12, pages 59–65, 2012.

[30] B. Jiang, M. Valstar, and M. Pantic. Action unit detection using sparse appearance descriptors in space-time video volumes. In *Automatic Face Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, pages 314–321, 2011.

[31] M. Jordan, Z. Ghahramani, T. Jaakkola, and L. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, 1999.

[32] S. Krishna, D. Colbry, J. Black, V. Balasubramanian, and S. Panchanathan. A Systematic Requirements Analysis and Development of an Assistive Device to Enhance the Social Interaction of People Who are Blind or Visually Impaired. In *Workshop on Computer Vision Applications for the Visually Impaired*, Marseille, France, Oct. 2008. James Coughlan and Roberto Manduchi.

[33] S. Kullback and R. A. Leibler. On information and sufficiency. *Ann. Math. Statist.*, 22(1):79–86, 1951.

[34] S. L and C. K. A temporal latent topic model for facial expression recognition. In *ACCV'11*.

[35] P. Lade, V. Balasubramanian, H. Venkateswara, and S. Panchanathan. Detection of changes in human affect dimensions using an adaptive temporal topic model. In *Multimedia and Expo (ICME), 2013 IEEE International Conference on*, 2013.

[36] P. Lade, V. Balasubramanian, N, and S. Panchanathan. Probabilistic topic models for huma affect analysis. In *NIPS Workshop on Topic Models*, 2013.

[37] T. S. Lee. Image representation using 2d gabor wavelets. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(10):959–971, Oct. 1996.

[38] P. Li, S. Phung, A. Bouzerdom, and F. H. C. Tivive. Feature selection for facial expression recognition. In *Visual Information Processing (EUVIP), 2010 2nd European Workshop on*, 2010.

[39] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[40] P. Lucey, J. 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. In *CVPRW*, 2010.

[41] Z. G. . P. M. Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):915–928, 2007.

[42] J. MacQueen. Some methods for classification and analysis of multivariate observations, 1967.

[43] T. McDaniel, S. Bala, J. Rosenthal, R. Tadayon, A. Tadayon, and S. Panchanathan. Affective haptics for enhancing access to social interactions for individuals who are blind. In *Universal Access in Human-Computer Interaction. Design and Development Methods for Universal Access*, pages 419–429. Springer, 2014.

[44] G. McLahlan and K. Basford. *Mixture Models: Inference and Applications to Clustering.* Marcel Dekker, 1988.

[45] P. Mermelstein. Distance measures for speech recognition, psychological and instrumental. *Pattern recognition and artificial intelligence*, 116:374–388, 1976.

[46] J. Nicolle, V. Rapp, K. Bailly, L. Prevost, and M. Chetouani. Robust continuous prediction of human emotions using multiscale dynamic cues. In *Proceedings of the 14th ACM International Conference on Multimodal Interaction*, 2012.

[47] T. Ojala, M. PietikÃinen, and D. Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1):51 – 59, 1996.

[48] D. Putthividhy, H. T. Attias, and S. S. Nagarajan. Topic regression multi-modal latent dirichlet allocation for image annotation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3408–3415. IEEE, 2010.

[49] A. Savran, H. Cao, M. Shah, A. Nenkova, and R. Verma. Combining video, audio and lexical indicators of affect in spontaneous conversation via particle filtering. ICMI '12.

[50] B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput.*, 10(5):1299–1319, July 1998.

[51] M. Schroder, E. Bevacqua, F. Eyben, H. Gunes, D. Heylen, M. ter Maat, S. Pammi, M. Pantic, C. Pelachaud, B. Schuller, E. de Sevin, M. Valstar, and M. Wollmer. A demonstration of audiovisual sensitive artificial listeners. In *Affective Computing and Intelligent Interaction and Workshops, 2009. ACII 2009. 3rd International Conference on*, pages 1–2, Sept 2009.

[52] B. Schuller, M. Valstar, R. Cowie, and M. Pantic. Avec 2012: The continuous audio/visual emotion challenge - an introduction. In *Proceedings of the 14th ACM International Conference on Multimodal Interaction*, 2012.

[53] L. Shang, K.-P. Chan, and G. Pan. Dttm: A discriminative temporal topic model for facial expression recognition. In *Advances in Visual Computing*. Springer, 2011.

[54] Y.-s. Shin. Recognizing facial expressions with pca and ica onto dimension of the emotion. In *Structural, Syntactic, and Statistical Pattern Recognition*, pages 916–922. Springer Berlin Heidelberg, 2006.

[55] K. Sikka, K. Dykstra, S. Sathyanarayana, G. Littlewort, and M. Bartlett. Multiple kernel learning for emotion recognition in the wild. In *Proceedings of the 15th ACM on International Conference on Multimodal Interaction*, ICMI '13, pages 517–524, 2013.

[56] M. Steyvers and T. Griffiths. *Latent Semantic Analysis: A Road to Meaning*, chapter Probabilistic Topic Models. Erlbaum, 2007.

[57] H. Tang and T. Huang. 3d facial expression recognition based on automatically selected features. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW '08. IEEE Computer Society Conference on*, 2008.

[58] A. N. Tikhonov. *Numerical methods for the solution of ill-posed problems*, volume 328. Springer Science & Business Media, 1995.

[59] M. Valstar and M. Pantic. Fully automatic recognition of the temporal phases of facial actions. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 42(1):28–43, 2012.

[60] M. Valstar, B. Schuller, K. Smith, T. Almaev, F. Eyben, J. Krajewski, R. Cowie, and M. Pantic. Avec 2014: 3d dimensional affect and depression recognition challenge. In *Proceedings of the 4th International Workshop on Audio/Visual Emotion Challenge*, pages 3–10, New York, NY, USA, 2014. ACM.

[61] M. Valstar, B. Schuller, K. Smith, F. Eyben, B. Jiang, S. Bilakhia, S. Schnieder, R. Cowie, and M. Pantic. Avec 2013: The continuous audio/visual emotion and depression recognition challenge. In *Proceedings of the 3rd ACM International Workshop on Audio/Visual Emotion Challenge*, 2013.

[62] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(2579-2605):85, 2008.

[63] J. Varadarajan, R. Emonet, and J.-M. Odobez. A sequential topic model for mining recurrent activities from long term video logs. *International Journal of Computer Vision*, 103(1):100–126, 2013.

[64] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511–I–518 vol.1, 2001.

[65] A. Wayne. Cusum based change point analysis. `http://www.variation.com/cpa/tech/changepoint.html`. Online; accessed January-2014.

[66] X. Xiong and F. De la Torre. Supervised descent method and its applications to face alignment. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[67] W. Zhang, S. Shan, W. Gao, X. Chen, and H. Zhang. Local gabor binary pattern histogram sequence (lgbphs): A novel non-statistical model for face representation and recognition. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 786–791, 2005.

[68] Z. Zhang and K. Raahemifar. Combined dictionary learning in facial expression recognition. *Journal of Signal and Information Processing*, 4(3B):86–90, 2013.

APPENDIX A

GIBBS SAMPLING FOR ATTM

In this Appendix the Gibbs sampling equations (6.5), (6.6), (6.2), (6.3) and (6.4) corresponding to the ATTM model discussed in Chapter 6 are derived in detail. The joint distribution of all the variables in ATTM is given by :

$$P(\boldsymbol{v},\boldsymbol{z},\boldsymbol{e^z}|\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{e^y}) = P(\boldsymbol{v}|\boldsymbol{z},\boldsymbol{\beta})P(\boldsymbol{z}|\boldsymbol{e^z},\boldsymbol{\alpha})P(\boldsymbol{e^z}|\boldsymbol{e^y}) \tag{A.1}$$

In order to derive the collapsed Gibbs sampling equations for $\boldsymbol{z}$ and $\boldsymbol{e^z}$, the three terms in the right hand side need to be derived.

The first term of Eq A.1, $P(\boldsymbol{v}|\boldsymbol{z},\boldsymbol{\beta})$ is given by:

$$\begin{aligned}
P(\boldsymbol{v}|\boldsymbol{z},\boldsymbol{\beta}) &= \int P(\boldsymbol{v}|\boldsymbol{z},\boldsymbol{\phi})P(\boldsymbol{\phi}|\boldsymbol{\beta})d\boldsymbol{\phi} \\
&= \prod_{k=1}^{K}\int\prod_{v=1}^{V}(\phi_{kv})^{n_k^v}\frac{1}{B(\boldsymbol{\beta})}\prod_{v=1}^{V}(\phi_{kv})^{\beta_v}d\boldsymbol{\phi_k} \\
&= \prod_{k=1}^{K}\int\frac{B(\boldsymbol{n_k}+\boldsymbol{\beta})}{B(\boldsymbol{\beta})}\underbrace{\frac{1}{B(\boldsymbol{n_k}+\boldsymbol{\beta})}\prod_{v=1}^{V}(\phi_{kv})^{n_k^v+\beta_v}d\boldsymbol{\phi_k}}_{Dir(\boldsymbol{\phi_k};\boldsymbol{n_k}+\boldsymbol{\beta})} \\
&= \prod_{k=1}^{K}\frac{B(\boldsymbol{n_k}+\boldsymbol{\beta})}{B(\boldsymbol{\beta})} \qquad\qquad (A.2)
\end{aligned}$$

The second term of Eq A.1, $P(\boldsymbol{z}|\boldsymbol{e^z},\boldsymbol{\alpha})$ is given by:

$$P(\boldsymbol{z}|\boldsymbol{e^z},\boldsymbol{\alpha}) = \prod_{t=1}^{T}\int P(\boldsymbol{z_t}|\boldsymbol{\theta_t})P(\boldsymbol{\theta_t}|\boldsymbol{e^z_t},\boldsymbol{\alpha})d\boldsymbol{\theta_t} \tag{A.3}$$

In the above equation $P(\boldsymbol{z_t}|\boldsymbol{\theta_t})$ is the multinomial distribution but to calculate the second term $P(\boldsymbol{\theta_t}|\boldsymbol{e^z_t},\boldsymbol{\alpha})$ we use $\boldsymbol{e^z_t}$ to transform the dirichlet distribution $P(\boldsymbol{\theta_t}|\boldsymbol{\alpha})$). As an example let $K = 5$ be the number of topics and $\boldsymbol{e^z_t} = (1,0,0,1,1)$, then $\boldsymbol{\theta_t} = (\theta_{t1},\theta_{t2},\theta_{t3},\theta_{t4},\theta_{t5})$ is transformed into $(\theta_{t1},\theta_{t4},\theta_{t2}+\theta_{t3}+\theta_{t5})$ i.e. $\boldsymbol{\theta_t}$ is tranformed to $\widehat{\boldsymbol{\theta}}_{\boldsymbol{t}} = (\boldsymbol{\theta_t^{k'}},(\sum_{k\notin\boldsymbol{k'}}\theta_{tk})+\theta_{tK})$ where K is the dummy topic and $\boldsymbol{k'}$ are the indices of topics to be retained in the current document. Using the aggregation property of Dirichlet distribution we have

$$\underbrace{(\boldsymbol{\theta_t^{k'}},(\sum_{k\notin\boldsymbol{k'}}\theta_{tk})+\theta_{tK})}_{\widehat{\boldsymbol{\theta}}_{\boldsymbol{t}}} \sim Dir(\widehat{\boldsymbol{\theta}}_{\boldsymbol{t}};\underbrace{(\boldsymbol{\alpha^{k'}},\overbrace{(\sum_{k\notin\boldsymbol{k'}}\alpha_k)+\alpha_K)}^{\alpha^{\neg\boldsymbol{k'}}}}_{\widehat{\boldsymbol{\alpha}}}) \tag{A.4}$$

and thus $P(\boldsymbol{\theta_t}|\boldsymbol{e^z_t},\boldsymbol{\alpha})$ is transformed to $P(\widehat{\boldsymbol{\theta}}_{\boldsymbol{t}}|\widehat{\boldsymbol{\alpha}})$. Expanding Eq A.2 we have,

$$\begin{aligned}
P(\boldsymbol{z}|\boldsymbol{e^z},\boldsymbol{\alpha}) &= \prod_{t=1}^{T}\int\prod_{k\in\boldsymbol{k'}}\theta_{tk}^{n_t^k}((\sum_{k\notin\boldsymbol{k'}}\theta_{tk})+\theta_{tK})^{n_t^K}\frac{\prod_{k\in\boldsymbol{k'}}\theta_{tk}^{\alpha_k-1}((\sum_{k\notin\boldsymbol{k'}}\theta_{tk})+\theta_{tK})^{\alpha^{\neg\boldsymbol{k'}}-1}}{B(\widehat{\boldsymbol{\alpha}})}d\boldsymbol{\theta_t} \\
&= \prod_{t=1}^{T}\frac{B(\boldsymbol{n_t^{k'}}+\boldsymbol{\alpha^{k'}},n_t^K+\alpha^{\neg\boldsymbol{k'}})}{B(\widehat{\boldsymbol{\alpha}})} \qquad\qquad (A.5)
\end{aligned}$$

The third term of Eq A.1, $P(\boldsymbol{e^z}|\boldsymbol{e^y})$ is given by:

$$P(\boldsymbol{e^z}|\boldsymbol{e^y}) = \prod_{t=2}^{T} \int P(\boldsymbol{e^z_t}|\boldsymbol{\delta^z_t}) P(\boldsymbol{\delta^z_t}|e^y_t, \boldsymbol{\theta_{t-1}}, \boldsymbol{\theta_t}) d\boldsymbol{\delta^z_t} \tag{A.6}$$

While $P(\boldsymbol{e^z_t}|\boldsymbol{\delta^z_t})$ is a product of K bernoulli distributions, $P(\boldsymbol{\delta^z_t}|e^y_t, \boldsymbol{\theta_{t-1}}, \boldsymbol{\theta_t})$ is calculated for two cases , $e^y_t = 0$ and $e^y_t = 1$.
Case 1 ($e^y_t = 0$):
If $e^y_t = 0$ then y_{t-1} is not similar to y_t and so uncommon topics have to be retained i.e. $\delta^z_{tk} \sim 1 \; \forall k \in$ uncommon topics and $\delta^z_{tk} \sim 0 \; \forall k \in$ common topics. Define $\alpha^z_{tk} = |\widehat{\theta}_{tk} - \widehat{\theta}_{t-1k}| + 0.0001$ and $\beta^z_{tk} = 100 - \alpha^z_{tk}$. In these calculations the use of 100 is not mandatory and any value can be used depending on the range to which θ values are scaled and in this work they have been scaled to lie between 0 and 100. 0.0001 has been used to represent an infinitesimal value to avoid $\alpha^z_{tk} = 0$.
Case 2 ($e^y_t = 1$):
If $e^y_t = 1$ then y_{t-1} and y_t are almost similar and thus common topics have to be retained i.e. $\delta^z_{tk} \sim 1 \; \forall k \in$ common topics and $\delta^z_{tk} \sim 0 \; \forall k \in$ uncommon topics. Define $\beta^z_{tk} = |\widehat{\theta}_{tk} - \widehat{\theta}_{t-1k}| + 0.0001$ and $\alpha^z_{tk} = 100 - \beta^z_{tk}$
Once α^z_{tk} and β^z_{tk} are calculated $\forall k$, we can estimate:

$$P(\boldsymbol{\delta^z_t}|e^y_t, \boldsymbol{\theta_{t-1}}, \boldsymbol{\theta_t}) = \prod_{k=1}^{K} \underbrace{P(\boldsymbol{\delta^z_{tk}}|\alpha^z_{tk}, \beta^z_{tk})}_{Beta(\delta^z_{tk};\alpha^z_{tk},\beta^z_{tk})}$$

where each of the probabilities in the product are calculated using Beta distributions. Using the above equation we can calculate $P(\boldsymbol{e^z}|\boldsymbol{e^y})$ as follows:

$$\begin{aligned} P(\boldsymbol{e^z}|\boldsymbol{e^y}) &= \prod_{t=2}^{T}\prod_{k=1}^{K} \int P(e^z_{tk}|\delta^z_{tk}) P(\delta^z_{tk}|\alpha^z_{tk}, \beta^z_{tk}) d\delta^z_{tk} \\ &= \prod_{t=2}^{T}\prod_{k=1}^{K} \int \delta_{tk^z}^{e^z_{tk}} (1-\delta^z_{tk})^{1-e^z_{tk}} \frac{\delta_{tk}^{\alpha^z_{tk}-1}(1-\delta^z_{tk})^{\beta^z_{tk}-1}}{B(\alpha_{tk}, \beta_{tk})} d\delta_{tk^z} \\ &= \prod_{t=2}^{T}\prod_{k=1}^{K} \int \frac{B(\alpha^z_{tk}+e^z_{tk}, \beta^z_{tk}-e^z_{tk}+1)}{B(\alpha^z_{tk}, \beta^z_{tk})} \underbrace{\frac{(\delta^z_{tk})^{\alpha^z_{tk}+e^z_{tk}-1}(1-\delta^z_{tk})^{\beta^z_{tk}-e^z_{tk}}}{B(\alpha^z_{tk}+e^z_{tk}, \beta^z_{tk}-e^z_{tk}+1)}}_{Beta(\delta^z_{tk};\alpha^z_{tk}+e^z_{tk},\beta^z_{tk}-e^z_{tk}+1))} d\delta^z_{tk} \\ &= \prod_{t=2}^{T}\prod_{k=1}^{K} \frac{B(\alpha^z_{tk}+e^z_{tk}, \beta^z_{tk}-e^z_{tk}+1)}{B(\alpha^z_{tk}, \beta^z_{tk})} \end{aligned} \tag{A.7}$$

The following sections contain the derivations for sampling $\boldsymbol{z}$ and $\boldsymbol{e^z}$ that correspond to the word and document levels respectively.

A.1 Sampling $\boldsymbol{z}$

Using collapsed Gibbs sampling the probability of sampling z_{tn} is conditioned on the rest of the topic assignments and is given below. For brevity α and β have been ignored. Let w be the n^{th} word in the document t, k is the topic assigned to this word.

$$\begin{aligned} P(z_{tn}=k|\boldsymbol{z}_{\neg(tn)},\boldsymbol{v},\boldsymbol{e^z},\boldsymbol{e^y}) &= \frac{P(\boldsymbol{v},\boldsymbol{z},\boldsymbol{e^z},\boldsymbol{e^y})}{P(\boldsymbol{v},\boldsymbol{z}_{\neg(tn)},\boldsymbol{e^z},\boldsymbol{e^y})} \\ &\propto \underbrace{\frac{P(\boldsymbol{v}|\boldsymbol{z})}{P(\boldsymbol{v}_{\neg(tn)}|\boldsymbol{z}_{\neg(tn)})}}_{a}\underbrace{\frac{P(\boldsymbol{z}|\boldsymbol{e^z})}{P(\boldsymbol{z}_{\neg(tn)}|\boldsymbol{e^z})}}_{b} \qquad \text{(A.8)} \end{aligned}$$

Evaluating the expression a of Eq A.8: Using Eq A.2 and retaining only topic k we have:

$$\begin{aligned} \frac{P(\boldsymbol{v}|\boldsymbol{z})}{P(\boldsymbol{v}_{\neg(tn)}|\boldsymbol{z}_{\neg(tn)})} &= \frac{B(\boldsymbol{n_k}+\boldsymbol{\beta})}{B(\boldsymbol{\beta})}\Big/\frac{B(\boldsymbol{n_{\neg k}}+\boldsymbol{\beta})}{B(\boldsymbol{\beta})} \\ &= \frac{\Gamma(n_k^w+\beta_w)}{\Gamma(n_{\neg k}^w+\beta_w)}\frac{\Gamma(\sum_{v\neq w}n_k^v+\sum_v\beta_v+n_{\neg k}^w)}{\Gamma(\sum_v n_k^v+\sum_v\beta_v)} \\ &\quad \text{where } n_k^w \text{ is no of times term } w \text{ has been assigned to topic } k \\ &\quad \text{and } n_{\neg k}^w \text{ is no of times term } w \text{ has been assigned to topic } k \\ &\quad \text{excluding current assignment, } \therefore n_{\neg k}^w = n_k^w - 1 \\ &= \frac{\Gamma(n_{\neg k}^w+1+\beta_w)}{\Gamma(n_{\neg k}^w+\beta_w)}\frac{\Gamma(\sum_{v\neq w}n_k^v+\sum_v\beta_v+n_{\neg k}^w)}{\Gamma(\sum_{v\neq w}n_k^v+\sum_v\beta_v+n_{\neg k}^w+1)} \\ &= \frac{n_{\neg k}^w+\beta_w}{\sum_{v\neq w}n_k^v+\sum_v\beta_v+n_{\neg k}^w} \qquad \text{(A.9)} \end{aligned}$$

Evaluating the expression b of Eq A.8: As mentioned earlier let w be the n^{th} word in the document t, k is the topic assigned to this word and $\boldsymbol{k'}$ is the set of topics retained in current document t. There are three cases to be considered, $e_{tk}^z = 1$ and $e_{tk}^z = 0$ depending on whether the current topic k is retained or not respectively and $k = K$ i.e. if the topic is the dummy topic.
Case 1($e_{tk}^z = 1$ i.e. k is retained):
Using Eq A.5 and the notation defined in Eq A.4 we have:

$$\begin{aligned} \frac{P(\boldsymbol{z}|\boldsymbol{e^z})}{P(\boldsymbol{z}_{\neg(tn)}|\boldsymbol{e^z})} &= \frac{B(\boldsymbol{n_t^{k'}}+\boldsymbol{\alpha^k},n_t^K+\alpha^{\neg\boldsymbol{k'}})}{B(\widehat{\boldsymbol{\alpha}})}\Big/\frac{B(\boldsymbol{n_{\neg t}^{k'}}+\boldsymbol{\alpha^{k'}},n_t^K+\alpha^{\neg\boldsymbol{k'}})}{B(\widehat{\boldsymbol{\alpha}})} \\ &= \frac{\Gamma(n_t^k+\alpha_k)}{\Gamma(n_{\neg t}^k+\alpha_k)}\frac{\Gamma(\sum_{i=1}^K\alpha_i+n_{\neg t})}{\Gamma(\sum_{i=1}^K\alpha_i+n_t)} \\ &\quad \text{(where } n_{\neg t} \text{ is the no of terms in } t \text{ without } w) \\ &\quad \text{(and } n_t \text{ is the no of terms in } t \text{ including } w) \\ &= \frac{n_{\neg t}^k+\alpha_k}{\sum_{i=1}^K\alpha_i+n_{\neg t}} \qquad \text{(A.10)} \end{aligned}$$

Case 2($e^z_{tk} = 0$ i.e. k is not retained):

$$\begin{aligned}
\frac{P(\boldsymbol{z}|\boldsymbol{e}^{\boldsymbol{z}})}{P(\boldsymbol{z}_{\neg(tn)}|\boldsymbol{e}^{\boldsymbol{z}})} &= \frac{\Gamma(\sum_{i=1}^{K}\alpha_i + n_{\neg t})}{\Gamma(\sum_{i=1}^{K}\alpha_i + n_t)} \\
&= \frac{1}{\sum_{i=1}^{K}\alpha_i + n_{\neg t}} \qquad \text{(A.11)}
\end{aligned}$$

Case 3($k = K$ i.e. k is the dummy topic):

$$\begin{aligned}
\frac{P(\boldsymbol{z}|\boldsymbol{e}^{\boldsymbol{z}})}{P(\boldsymbol{z}_{\neg(tn)}|\boldsymbol{e}^{\boldsymbol{z}})} &= \frac{\Gamma(n_t^K + \sum_{i\notin \boldsymbol{k'}}\alpha_i + \alpha_K)}{\Gamma(n_{\neg t}^K + \sum_{i\notin \boldsymbol{k'}}\alpha_i)} \frac{\Gamma(\sum_{i=1}^{K}\alpha_i + n_{\neg t})}{\Gamma(\sum_{i=1}^{K}\alpha_i + n_t)} \\
&= \frac{n_{\neg t}^K + \sum_{i\notin \boldsymbol{k'}}\alpha_i + \alpha_K}{\sum_{i=1}^{K}\alpha_i + n_{\neg t}} \qquad \text{(A.12)}
\end{aligned}$$

Substituting all the above results in the Eq A.8 and considering the 3 cases discussed above, we have the following sampling equations for z_{tn}:
Case 1: $e_{tk} = 0$ and $k \neq K$
Substituting Eqs A.9 and A.11 for expressions a and b respectively in Eq A.8 we have:

$$P(z_{tn} = k|\boldsymbol{z}_{\neg(tn)}, \boldsymbol{v}, \boldsymbol{e}^{\boldsymbol{z}}, \boldsymbol{e}^{\boldsymbol{y}}) = \frac{n_{\neg k}^w + \beta_w}{\sum\limits_{v\neq w} n_k^v + \sum\limits_{v}\beta_v + n_{\neg k}^w} \frac{1}{\sum\limits_{i=1}^{K}\alpha_i + n_{\neg t}} \qquad \text{(A.13)}$$

Case 2: $e_{tk} = 1$ and $k \neq K$
Substituting Eqs A.9 and A.10 for expressions a and b respectively in Eq A.8 we have:

$$P(z_{tn} = k|\boldsymbol{z}_{\neg(tn)}, \boldsymbol{v}, \boldsymbol{e}^{\boldsymbol{z}}, \boldsymbol{e}^{\boldsymbol{y}}) = \frac{n_{\neg k}^w + \beta_w}{\sum\limits_{v\neq w} n_k^v + \sum\limits_{v}\beta_v + n_{\neg k}^w} \frac{n_{\neg t}^k + \alpha_k}{\sum\limits_{i=1}^{K}\alpha_i + n_{\neg t}} \qquad \text{(A.14)}$$

Case 3: $k = K$
Substituting Eqs A.9 and A.12 for expressions a and b respectively in Eq A.8 we have:

$$P(z_{tn} = K|\boldsymbol{z}_{\neg(tn)}, \boldsymbol{v}, \boldsymbol{e}^{\boldsymbol{z}}, \boldsymbol{e}^{\boldsymbol{y}}) = \frac{n_{\neg K}^w + \beta_w}{\sum\limits_{v\neq w} n_K^v + \sum\limits_{v}\beta_v + n_{\neg K}^w} \frac{n_{\neg t}^K + \sum\limits_{i\notin \boldsymbol{k'}}\alpha_i + \alpha_K}{\sum\limits_{i=1}^{K}\alpha_i + n_{\neg t}} \qquad \text{(A.15)}$$

A.2 Sampling $\boldsymbol{e}^{\boldsymbol{z}}$

As described in the previous section, we again use collapsed Gibbs sampling to evaluate the probability of the random variable e^z_{tk} as described below.

$$\begin{aligned}
P(e^z_{tk}|\boldsymbol{e}^{\boldsymbol{z}}{}_{\neg tk}, \boldsymbol{z}, \boldsymbol{v}, \boldsymbol{e_y}) &= \frac{P(\boldsymbol{v}, \boldsymbol{z}, \boldsymbol{e}^{\boldsymbol{z}}, \boldsymbol{e}^{\boldsymbol{y}})}{P(\boldsymbol{v}, \boldsymbol{z}, \boldsymbol{e}^{\boldsymbol{z}}{}_{\neg tk}, \boldsymbol{e}^{\boldsymbol{y}})} \\
&\propto \underbrace{\frac{P(\boldsymbol{z}|\boldsymbol{e}^{\boldsymbol{z}})}{P(\boldsymbol{z}_{\neg tk}|\boldsymbol{e}^{\boldsymbol{z}}{}_{\neg tk})}}_{a} \underbrace{\frac{P(\boldsymbol{e}^{\boldsymbol{z}}|\boldsymbol{e_y})}{P(\boldsymbol{e}^{\boldsymbol{z}}{}_{\neg tk}|\boldsymbol{e_y})}}_{b} \qquad \text{(A.16)}
\end{aligned}$$

where $\boldsymbol{z}_{\neg tk}$ is the set of all topic assignments in document t except those that have been assigned topic k.

Evaluating the expression a of Eq A.17:

Case 1: $e^z_{tk} = 1$ Expanding the formula in Eq A.10 we have

$$
\begin{aligned}
\frac{P(\boldsymbol{z}|\boldsymbol{e^z})}{P(\boldsymbol{z}_{\neg tk}|\boldsymbol{e^z}_{\neg tk})} &= \frac{\Gamma(n^k_t + \alpha_k)}{\Gamma(\alpha_k)} \frac{\Gamma(\sum\limits_{i=1}^{K} \alpha_i + n_{\neg t})}{\Gamma(\sum\limits_{i=1}^{K} \alpha_i + n_t)} \\
&\quad \text{(where } n_{\neg t} \text{ is the no of terms in } t \text{ excluding those assigned to } k\text{)} \\
&= \frac{\prod_{s=0}^{n^k_t} (\alpha_k + s)}{\prod_{s=n_{\neg t}}^{n_t} (\sum\limits_{i=1}^{K} \alpha_i + s)} \qquad \text{(A.17)} \\
& \qquad \text{(A.18)}
\end{aligned}
$$

where $\Gamma(a+b) = \left(\prod_{i=0}^{b}(a+i)\right)\Gamma(a)$ is used to derive the above equation. *Case 2*: $e^z_{tk} = 0$ Again expanding the formula in Eq A.10 we have

$$
\begin{aligned}
\frac{P(\boldsymbol{z}|\boldsymbol{e^z})}{P(\boldsymbol{z}_{\neg tk}|\boldsymbol{e^z}_{\neg tk})} &= \frac{\Gamma(\sum\limits_{i=1}^{K} \alpha_i + n_{\neg t})}{\Gamma(\sum\limits_{i=1}^{K} \alpha_i + n_t)} \\
&= \frac{1}{\prod_{s=n_{\neg t}}^{n_t} (\sum\limits_{i=1}^{K} \alpha_i + s)} \qquad \text{(A.19)}
\end{aligned}
$$

Evaluating the expression b of Eq A.17:

Using the Eq A.7 and ignoring all terms except for those that correspond to t and k, the expression b becomes:

$$
\begin{aligned}
\frac{P(\boldsymbol{e^z}|\boldsymbol{e_y})}{P(\boldsymbol{e^z}_{\neg tk}|\boldsymbol{e_y})} &\propto \frac{B(\alpha^z_{tk} + e^z_{tk}, \beta^z_{tk} - e^z_{tk} + 1)}{B(\alpha^z_{tk}, \beta^z_{tk})} \\
&= \frac{\Gamma(\alpha^z_{tk} + e^z_{tk})\Gamma(\beta^z_{tk} - e^z_{tk} + 1)}{\Gamma(\alpha^z_{tk} + \beta^z_{tk} + 1)} \frac{\Gamma(\alpha^z_{tk} + \beta^z_{tk})}{\Gamma(\alpha^z_{tk})\Gamma(\beta^z_{tk})}
\end{aligned}
$$

Case 1: $e^z_{tk} = 1$

$$
\begin{aligned}
\frac{P(\boldsymbol{e^z}|\boldsymbol{e_y})}{P(\boldsymbol{e^z}_{\neg tk}|\boldsymbol{e_y})} &= \frac{\Gamma(\alpha^z_{tk} + 1)\Gamma(\beta^z_{tk})}{\Gamma(\alpha^z_{tk} + \beta^z_{tk} + 1)} \frac{\Gamma(\alpha^z_{tk} + \beta^z_{tk})}{\Gamma(\alpha^z_{tk})\Gamma(\beta^z_{tk})} \\
&= \frac{\alpha^z_{tk}}{\alpha^z_{tk} + \beta^z_{tk}} \qquad \text{(A.20)}
\end{aligned}
$$

Case 2: $e^z_{tk} = 0$

$$
\frac{P(\boldsymbol{e^z}|\boldsymbol{e_y})}{P(\boldsymbol{e^z}_{\neg tk}|\boldsymbol{e_y})} = \frac{\beta^z_{tk}}{\alpha^z_{tk} + \beta^z_{tk}} \qquad \text{(A.21)}
$$

By substituting Eqs A.17 A.19 A.20 and A.21 in Eq A.17 we derive the following four cases:
Case 1: $e^z_{tk} = 1$ and $e^y_t = 0$ Substituting Eqs A.17 and A.20 in Eq A.17 we have:

$$P(e^z_{tk}|\boldsymbol{e^z}_{\neg tk}, \boldsymbol{z}, \boldsymbol{v}, \boldsymbol{e_y}) = \frac{\prod_{s=0}^{n_t^k}(\alpha_k + s)}{\prod_{s=n_{\neg t}}^{n_t}(\sum_{i=1}^{K} \alpha_i + s)} \frac{\alpha^z_{tk}}{\alpha^z_{tk} + \beta^z_{tk}} \tag{A.22}$$

$$\text{where } \alpha^z_{tk} = |\widehat{\theta}_{tk} - \widehat{\theta}_{t-1k}| + 0.0001 \text{ and } \beta^z_{tk} = 100 - \alpha^z_{tk}$$

Case 2: $e^z_{tk} = 1$ and $e^y_t = 1$

$$P(e^z_{tk}|\boldsymbol{e^z}_{\neg tk}, \boldsymbol{z}, \boldsymbol{v}, \boldsymbol{e_y}) = \frac{\prod_{s=0}^{n_t^k}(\alpha_k + s)}{\prod_{s=n_{\neg t}}^{n_t}(\sum_{i=1}^{K} \alpha_i + s)} \frac{\alpha^z_{tk}}{\alpha^z_{tk} + \beta^z_{tk}} \tag{A.23}$$

$$\text{where } \beta^z_{tk} = |\widehat{\theta}_{tk} - \widehat{\theta}_{t-1k}| + 0.0001 \text{ and } \alpha^z_{tk} = 100 - \beta^z_{tk}$$

Case 3: $e^z_{tk} = 0$ and $e^y_t = 0$ Substituting Eqs A.19 and A.21 in Eq A.17 we have:

$$P(e^z_{tk}|\boldsymbol{e^z}_{\neg tk}, \boldsymbol{z}, \boldsymbol{v}, \boldsymbol{e_y}) = \frac{1}{\prod_{s=n_{\neg t}}^{n_t}(\sum_{i=1}^{K} \alpha_i + s)} \frac{\beta^z_{tk}}{\alpha^z_{tk} + \beta^z_{tk}} \tag{A.24}$$

$$\text{where } \alpha^z_{tk} = |\widehat{\theta}_{tk} - \widehat{\theta}_{t-1k}| + 0.0001 \text{ and } \beta^z_{tk} = 100 - \alpha^z_{tk}$$

Case 4: $e^z_{tk} = 0$ and $e^y_t = 1$

$$P(e^z_{tk}|\boldsymbol{e^z}_{\neg tk}, \boldsymbol{z}, \boldsymbol{v}, \boldsymbol{e_y}) = \frac{1}{\prod_{s=n_{\neg t}}^{n_t}(\sum_{i=1}^{K} \alpha_i + s)} \frac{\beta^z_{tk}}{\alpha^z_{tk} + \beta^z_{tk}} \tag{A.25}$$

$$\text{where } \beta^z_{tk} = |\widehat{\theta}_{tk} - \widehat{\theta}_{t-1k}| + 0.0001 \text{ and } \alpha^z_{tk} = 100 - \beta^z_{tk}$$

APPENDIX B

CODE SNIPPETS

Listing B.1: Code to calculate the Variational LowerBound

```
   double compute_likelihood(int doc, HashMap<Integer,Integer> currDoc,
       double[][] phi){
           double likelihood = 0, digSum = 0, varGammaSum = 0;
      double[] dig=new double[K];
      int total= currDoc.size();
      int[] termsInDoc= new int[total];
      int[] counts= new int[total];
      int index=0;
      for(Integer key: currDoc.keySet()){
         termsInDoc[index]= key;
         counts[index]= currDoc.get(key);
         index++;
      }
      for (int k = 0; k < K; k++){
         dig[k] = Utilities.digamma(varGamma[doc][k]);
         varGammaSum += varGamma[doc][k];
      }
      digsum = Utilities.digamma(var_gamma_sum);
      likelihood = Utilities.LogGamma(alpha * K)- K *
          Utilities.LogGamma(alpha)- (Utilities.LogGamma(varGammaSum));
      for (int k = 0; k < K; k++){
         likelihood += (alpha - 1)*(dig[k] - digSum) +
             Utilities.LogGamma(varGamma[doc][k])
               - (varGamma[doc][k] - 1)*(dig[k] - digSum);
         for (int n = 0; n < total; n++){
            if (phi[n][k] > 0){
               likelihood += counts[n]*(phi[n][k]*((dig[k] - digSum) -
                   Math.log(phi[n][k])+ Math.log(pi[k][termsInDoc[n]])));
            }
         }
      }
      return(likelihood);
   }
```

Listing B.2: Map Reduce Code to run the Variational E step in parallel

```
ExecutorService service = Executors.newFixedThreadPool(threads);
List<Future<EStepOutput>> futures = new ArrayList<Future<EStepOutput>>();
// Map
int numThreads= Runtime.getRuntime().availableProcessors();
for(int i=0; i<numThreads;i++){
   // Mapper for document t + i
   Callable<EStepOutput> mapper= this.new EStepMapper(t+i,i);
   futures.add(service.submit(mapper));
}
// Reduce
for(Future<EStepOutput> futureoutput: futures){
   likelihood+= futureoutput.get().likelihood;
}
```

```
t += numThreads;
after= System.currentTimeMillis();
service.shutdown();
```

Listing B.3: Reusable Calculation of Log Normal probabilities in M-Step of DGMM

```
for (int k = 0; k < K; k++){ // K Topics
   for (int n = 0; n < N; n++){ // N features
   double total=0;
   for(int t=0; t<documents.length; t++){ // all documents
      log_prob_w[k][n][t]= Utilities.normalPDF(documents[t][n],
          topicMeans[k][n], Math.sqrt(topicSigmas[k][n]));
      total+= log_prob_w[k][n][t];
   }
   for(int t=0; t<documents.length; t++){
      log_prob_w[k][n][t]= log_prob_w[k][n][t]/total;
      log_prob_w[k][n][t]= Math.log(log_prob_w[k][n][t]);
      if(Double.isNaN(log_prob_w[k][n][t]) ||
          Double.isInfinite(log_prob_w[k][n][t]))
         log_prob_w[k][n][t] = -1000;
   }
   }
}
```

Listing B.4: Java Code that computes unsupervised and supervised likelihoods for SLDA

```
   double[] compute_likelihood(int doc, HashMap<Integer,Integer> currDoc,
       double[][] phi, double[][] EZTZ){
      double likelihoodW = 0, likelihoodY = 0, digsum = 0, varGammaSum = 0;
      // likelihoodW: likelihood terms from SLDA that does not contain ys
      // likelihoodY: likelihood terms from SLDA that contain ys
      double[] dig=new double[K]; int total= currDoc.size();
      int[] termsInDoc= new int[total]; int[] counts= new int[total];
      int index=0;
      for(Integer key: currDoc.keySet()){
         termsInDoc[index]= key;
         counts[index]= currDoc.get(key);
         index++;
      }
      likelihoodY += Math.log(1/(Math.sqrt(2*Math.PI)*delta))-
          (Math.pow(ys[doc],2)/2);
      double term1=0; double[] EZ= new double[K];
      for (int k = 0; k < K; k++){
         EZ[k]=0;
         for (int n = 0; n < total; n++){
            EZ[k]+= counts[n]*phi[n][k];
         }
         EZ[k]/=docLengths[doc];
         term1+= b[k]*EZ[k]*ys[doc];
```

```
  }
  double term2=0; Matrix result = new Matrix(EZTZ);
  term2= new Matrix(b,K).transpose().times(result).times(new
     Matrix(b,K)).getArray()[0][0];
  term2/=(2*docLengths[doc]*docLengths[doc]);
  likelihoodY += (term1-term2)/delta;
  for (int k = 0; k < K; k++){
    dig[k] = Utilities.digamma(varGamma[doc][k]);
    varGammaSum += varGamma[doc][k];
  }
  digsum = Utilities.digamma(varGammaSum);
  likelihoodW = Utilities.LogGamma(alpha * K)- K *
     Utilities.LogGamma(alpha)- (Utilities.LogGamma(varGammaSum));
  for (int k = 0; k < K; k++){
    likelihoodW += (alpha - 1)*(dig[k] - digsum) +
       Utilities.LogGamma(varGamma[doc][k]) - (varGamma[doc][k] -
       1)*(dig[k] - digsum);
    for (int n = 0; n < total; n++){
      if (phi[n][k] > 0){
        likelihood_w += counts[n]*(phi[n][k]*((dig[k] - digsum) -
           Math.log(phi[n][k]) + log(pi[k][termsInDoc[n]])));
      }
    }
  }
  double[] wyLikelihoods = new double[2];
  wyLikelihoods[0] = likelihoodW;
  wyLikelihoods[1] = likelihoodY;
  return(wyLikelihoods);
}
```