Major Project Report

On

SEMI-SUPERVISED VIDEO SEMANTIC RECOGNITION

Submitted by

12IT79 Siddharth P. Ramakrishna

12IT09 Anuj Kumar

12IT62 Rohit Kumar

Under the Guidance of,

Mr.Dinesh Naik

Department of Information Technology, NITK Surathkal

Date of Submission: 16/11/2015

Department of Information Technology

National Institute of Technology Karnataka, Surathkal

November 2015

Department of Information Technology, NITK Surathkal
Major Project
End Semester Evaluation Report (November 2015)

Course Code : IT 449

Course Title: Major Project - 1

Project Title: *SEMI-SUPERVISED VIDEO SEMANTIC RECOGNITION*

Project Group:

Name of the Student	Register No.	Signature with Date
Anuj Kumar	12IT09	
Rohit Kumar	12IT62	
Siddharth P	12IT79	

Place:Surathkal

Date:16th November, 2015 *(Mr.Dinesh Naik)*

Abstract

The accuracy and efficiency of the Video Semantic Recognition can be improved by performing feature selection on the features that has been extracted from the video. A subset of features has been selected. The feature set is highly dimensional because a video is generally very rich in semantic. Feature selection is done for a better, compact and accurate representation of the video. Since it is difficult and time consuming to compute all the labeled videos and do a human annotated classification, the number of labeled videos is small. Nowadays, most of the applications have a large amount of unlabeled videos. And supervised feature selection will fail to realise the important features since the labelled videos are less and therefore it is difficult to do a classification to target classes. So this gives an intuition to use a mixture of labelled and unlabelled videos and apply semi supervised algorithms to select relevant features from the video such that they are discriminative to respective target classes by using effectively the information associated with the large amount of unlabeled videos. In this project, we will provide a comparative study of which algorithm has the most efficiency and accuracy under the given assumptions. Following are the three semi supervised algorithms that we will be approaching for the study Spline Regression, Graph Based Semi Supervised Algorithm and Adaboost. In the experiments to come, a comparative study will be provided on the three typical tasks of video semantic recognition, namely Video Concept Detection, Video Classification and Human Action Recognition.

Keywords: *Regression, Video Classification, Adaboost*

Contents

1 **Introduction** 1

2 **Literature Survey** 3

2.1 ***Background*** . . . 3

2.2 ***Outcome of Literature Survey*** . . . 6

2.3 ***Problem Statement*** . . . 6

2.4 ***Objectives*** . . . 6

3 **Research Methodology** 7

3.1 ***Framework Of The Project*** . . . 7

3.2 ***Getting the key frames and performing feature extraction*** . . . 7

3.2.1 ***Data Sets under consideration:*** . . . 7

3.2.2 ***KeyFrame Extraction :*** . . . 8

3.2.3 ***Feature Extraction :*** . . . 8

3.3 ***Spline Regression*** . . . 15

3.4 ***Graph Based Semi Supervised Algorithm*** . . . 19

3.5 ***AdaBoost Algorithm*** . . . 20

3.6 ***Performance Analysis :*** . . . 23

4 **Work Done and Results** 24

5 **Conclusion and Future Work** 27

6 **Time Line Of The Project** 28

7 **References** 29

List of Figures

3.1 Framework Diagram 7
3.2 Key Frames From Nature 9
3.3 Key Frames From Walk 9
3.4 Key Frames From Building 9
3.5 Blue Histogram 12
3.6 Green Histogram 12
3.7 Red Histogram 12
3.8 Reference Image 14
3.9 Blur Red Image 15
3.10 Thresholding of the Image 15
3.11 Semi Supervised Feature Selection Via Spline Regression 16
3.12 Supervised Feature Selection Via Spline Regression 17
3.13 Supervised Feature Selection Via Spline Regression Interpolation 18
3.14 AdaBoost Algorithm 22
4.1 Dimensions for first 6 frames for Video 0 25
4.2 Feature Matrix For One Frame 25
4.3 Feature Matrix For One Video 25
4.4 Predicted Unlabeled Instances 26

1 Introduction

Most of the applications of video semantic recognition has data represented by high dimensional feature vectors. It has been observed that one can extract high dimensional visual features which are heterogeneous from a given video frame. Features such as global like direction of edges, gabor, and color moment and some local features such as space time interest points. In this high dimensional space of observed features, it is difficult to classify video samples of different classes and is called as curse of dimensionality problem. And since it is known that more the number of irrelevant features, the degree of polynomial tends to increase and thus the process of training used for classification tends to be overfitting. So, this issue has to be resolved.

One solution to this issue is dimensionality reduction. The original high dimensional feature space is reduced onto a new reduced dimensionality feature space and the video samples are being represented in that new space. The two basic methods of mapping in dimensionality reduction or the other way round is done either by constructing new features or by keeping a subset of the features that has been obtained from the original space. For the former mapping, two major strands are PCA(The Principal Component Analysis) and the Isometric Mapping of data manifolds. PCA uses the linear subspace learning and the latter uses the nonlinear manifold learning methods. In this project, the dimensionality reduction approach has been explored, making a subset of the features from the original space, and its possible applications to the video semantic recognition.

The common approach of dimensionality reduction, feature selection, has an important role in improving the accuracy and efficieny of analysis of the video samples. Firstly, the computations has been redduced as the new dimension of the feature subset is very low. And then the noisy ones are being removed for a efficient representation of the video samples, thus making a better classification result.

Algorithms to perform feature selection can be basically classified into the following groups, supervised feature selection and the unsupervised feature selection. In supervised features selection, the algorithm computes the relevance of the feature by evaluating the correlation of that feature with the target classes. Pastly, Fischer score, robust regression and regression algorithms like sparse multi output regression has been used to select the features based on the labels on the training data. Since the discriminative information is there enclosed with the labels, the supervised feature selection algorithm is able to select

the discriminative features from the training data. The unsupervised feature selection algorithm uses the separability and the variance among the data to evaluate feature relevance where there are no labels in the training data.

The usual practice criteria that has been observed is to select some of the features that preserve the distribution of the data and structure of the data set from the entire feature set efficiently. But the difficulty here is that the label information sometimes is not directly available and therefore the unsupervised feature selection algorithm has more problems in selecting the discriminitating features using the distribution and the structure of the data.

It has been observed that the amount of unlabeled videos are huge and getting large amount of high quality labelled videos is very difficult. Since the amount of labelled videos is fairly small, supervised feature selection algorithm will fail to find out the features that are relevant enough to discrimate them to respective target classes. Therefore a need for semi supervised feature selection algorithm approach could be used to identify the most relevant features that are discriminative enough to target classes. In order to make use of the labelled and the unlabelled video samples, the semi superised feature selection algorithm makes use of the distribution and the structure of the data to evaluate the relevance of the features. Several semi supervised feature selection algorithms based on the spectral assumption, the geometry of distribution of data, and others.

So, in this project, we propose a comparative study for semi supervised feature selection via the following semi supervised algorithms like the Spline regression, the graph based semi supervised algorithm and the adaboost algorithm.

In the experiments to come, standard benchmark video samples as training data sets are used to provide a comparative study on the performance of video semantic recognition by semi supervised feature selection algorithms which basically corresponds to the three important video semantic recognition taks. A comparative study on the performance of video semantic recognition using the semi supervised feature selection algorithms will be provided.

2 Literature Survey

2.1 *Background*

Semi Supervised classification is a special form of classification. Traditional classifiers use only labeled data (feature / label pairs) to train. Labeled instances however are often difficult, expensive, or time consuming to obtain, as they require the efforts of experienced human annotators. Meanwhile unlabeled data may be relatively easy to collect, but there has been few ways to use them. Semi-supervised learning addresses this problem by using large amount of unlabeled data, together with the labeled data, to build better classifiers. Because semi-supervised learning requires less human effort and gives higher accuracy, it is of great interest both in theory and in practice. Labels are hard to obtain while unlabeled data are abundant, therefore semi-supervised learning is a good idea to reduce human labor and improve accuracy. Anecdotally, the fact that unlabeled data do not always help semi-supervised learning has been observed by multiple researchers. For example people have long realized that training Hidden Markov Model with unlabeled data (the Baum-Welsh algorithm, which by the way qualifies as semi-supervised learning on sequences) can reduce accuracy under certain initial conditions (Elworthy, 1994).

Many semi supervised learning methods are there. Some often-used methods include: EM with generative mixture models, self-training, co-training, transductive support vector machines, and graph-based methods.

There is no direct answer to the question which method suits the best. Because labeled data is scarce, semi- supervised learning methods make strong model assumptions. Ideally one should use a method whose assumptions fit the problem structure. This may be difficult in reality. Nonetheless we can try the following checklist: Do the classes produce well clustered data? If yes, EM with generative mixture models may be a good choice; Do the features naturally split into two sets? If yes, co-training may be appropriate; Is it true that two points with similar features tend to be in the same class? If yes, graph-based methods can be used; Already using SVM? Transductive SVM is a natural extension; Is the existing supervised classifier complicated and hard to modify? Self-training is a practical wrapper method.

Semi-supervised learning methods use unlabeled data to either modify or reprioritize hypotheses obtained from labeled data alone.

Transductive learning will be used to contrast inductive learning. A learner is transductive if it only works on the labeled and unlabeled training data, and cannot handle unseen data. The early graph-based methods are often transductive. Inductive learners can naturally handle unseen data. People sometimes use the analogy that transductive learning is take-home exam, while inductive learning is in-class exam.

.In order to improve the accuracy and efficiency for the video semantic recognition, one can perform feature selection on the video features extracted to choose a subset of the features from the set of high dimensional features to have a better, compact and accurate video presentation. Yang Yan (2014) developed an iterative algorithm for the experiment and were able to prove its convergency. In the experiments, the three important and typical tasks of semantic recognition in videos, i.e, video classification, human action recognition and video concept detection, are used to demonstrate that the proposed algorithm semi supervised feature selection via spline regression achieves good performance when compared with the other state-of-the-art methods.

The automatic understanding of the audio and the visual content for the multi media retrieval is not an easy task, because the meaning basically the appearance of a certain event or a concept is strongly determined by the contextual information present. Ewerth Bernd (2007) show that it is quite possible to adaptively grasp and learn the appearance of some certain objects or some events for a particular video (test video) using unlabeled data so that it can improve over subsequent retrieval process. So, basically first an initial model is obtained using supervised learning making use of a set of appropriate training videos. Then, this model is being used to rank the shots for each test video Vi separately. This ranking system is used to label the most important and the least important shots in the video Vi for further use as training or the input data in a semi supervised learning process. So basically using these labeled training data, important features are being selected for the concept in consideration for the video Vi. Then, the two additional classifiers are being trained on the data of this video which are automatically labeled. Adaboost and SVM (Support Vector Machines) are being incorporated for the feature selection process and the classification ensemble process. And then finally the newly obtained trained classifiers and initial model act as an ensemble. They performed experiments on the TRECVID 2005 video data and demonstrated the feasibility of proposed

learning scheme for the certain high level concepts.

The multimedia applications require video management and it is becoming more and more desirable and important to give proper video data index techniques which are capable of representing the semantics in video data that are rich in nature. In most real-time applications, there is a need for efficiently query processing which is one of the other for the use of such techniques. Serhan Khatib(2000) have presented the models that can use the object motion information to characterize the further events that can allow further subsequent retrieval. Different algorithms for different search in terms of spatiotemporal cases in temporal and spatials translation and scaling invariance have been in development using various image processing and signal processing techniques. PICTURESQUE (pictorial information and content transformation unified retrieval engine for spatiotemporal queries) to basically prove the methods that has been proposed. With the development of such technologies, it will enable true multimedia search engines which will allow searching of the video data in digital form and indexing based on the true content.

The problem for the graph classification has attracted great interest in recent times. Currently going research assumes that there is large number of labeled training graphs available. But most of the times labels of the graph data are really very expensive and are difficult to obtain, and at the same there are large amounts of unlabeled graph data present. Kong Yu (2010) developed the algorithm whose aim is to efficiently search for the optimal sub graph features containing both the labeled and the un labeled graphs.

The accuracy and efficiency of the Video Semantic Recognition can be improved by performing feature selection on the features that has been extracted from the video. A subset of features has been selected. The feature set is highly dimensional because a video is generally very rich in semantic. Han Yang (2014) show that Feature selection is done for a better, compact and accurate representation of the video. Since it is difficult and time consuming to compute all the labeled videos and do a classification, the number of labelled videos is small.

2.2 *Outcome of Literature Survey*

Semi supervised classification algorithms are developed to overcome the disadvantages due to supervised and unsupervised classification algorithms and provide a concrete study of the algorithms in the semi supervised domain to utilize them to provide a better video semantic recognition based on the assumptions made on the type of data and constraints.

2.3 *Problem Statement*

Given a large set of labelled and unlabelled training videos where num(labelled) << num(unlabelled), apply semi supervised learning techniques to train them and subsequently classify test videos

2.4 *Objectives*

1. Getting the key frames and performing feature extraction
2. Feature Reduction using Spline Regression
3. Implementing the graph based semi supervised learning algorithm using Gaussian fields and harmonic functions
4. Implementing the AdaBoost Semi Supervised algorithm and comparing the performance with the Graph Based Semi Supervised Learning algorithm
5. Compare results and plot graphs to determine the quality of the algorithm with Varying Heuristics and Assumptions in our data set

3 Research Methodology

3.1 *Framework Of The Project*

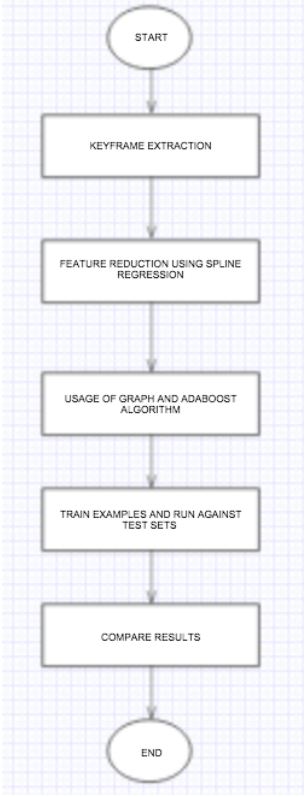


Figure 3.1: Framework Diagram

Figure 3.1 shows the pipeline of the project. Video datasets are collected and the key frames has been extracted from the videos and then key features has been extracted to construct a feature matrix. Then feature reduction will be done using spline regression and using graph based semi-supervised algorithm and AdaBoost Algorithm, finally the Videos are classified.

3.2 *Getting the key frames and performing feature extraction*

3.2.1 *Data Sets under consideration:*

In order to implement the project, three different categories of video data have been selected. They are as follows:

1. Human Activity Data

2. Building data

3. Nature data

The rationale behind choosing these classes is the underlying fact that they can be more easily distingushable as opposed to 3 highly co relating datasets, say for example, human walking and human running. In order to classify these type of videos, a large amount of complex high level feature extraction is required. However with these three(and more in the future), we can effectively classify them by dealing with more low level features which are a lot more simple to obtain, provide a higher degree of accuracy and can be used more effectively in finding high level features if required. In total we have 33 videos-12-human, 11-building and 11 nature. Figure 3.1 have some sample frames of nature. Figure 3.2 have some sample frames of human actions. Figure 3.3 have some sample frames of building.

The videos are in .avi format and are range from around 6 seconds to 22 seconds. The first step with the data set is to obtain keyframes.With the help of OpenCv methods, we can extract keyframes for a given video which are stored in .png format. However, given a video of 6 seconds duration, it was found that around 150 keyframes were generated, many of them redundant. In order to get around this task of reducing key frames, the following method was adopted.

3.2.2 *KeyFrame Extraction :*

Filmstrip.py : Filmstrip is an OpenCV/Python based set of scripts for extracting keyframes from video. It was written to extract the data that powers the openvisconf videos visualization. We have utilised the frame skipping feature of this script.
Frame skipping: Reducing the number of frames in the video by extracting one frame per second. The videos are passed to this algorithm which does the task of fitting all the frames within 1s. With the help of this, we can use openCV's key frame extraction methods to work within a shorter period of time and as a result, obtain lesser number of frames. It was found that the number of frames per video were ranging between 16 and 28 which is clearly, much lesser than 150 per video and a large number of redundant frames are removed which saves a lot of unnecessary processing time. The images are of dimensions 120*160.

(a) Nature Key Frame-1

(b) Nature Key Frame-2

(c) Nature Key Frame-3

Figure 3.2: Key Frames From Nature

(a) Walk Key Frame-1

(b) Walk Key Frame-2

(c) Walk Key Frame-3

Figure 3.3: Key Frames From Walk

(a) Building Key Frame-1

(b) Building Key Frame-2

(c) Building Key Frame-3

Figure 3.4: Key Frames From Building

3.2.3 *Feature Extraction :*

This step is highly crucial to the functioning of the project. The goal of feature extraction is to end up with a feature vector for each keyframe and therefore, a video can be represented by many of these feature vectors. Naturally, our matrix is created in this manner. It is an n*f*v matrix, where n is for the number of videos, f is for the number of key frames corresponding to the video and v is the dimension of the feature vector corresponding to each frame. As of now, n = 34, f =30 and v = 9. Our main area of interest here is to determine these 9 elements in our feature vector. They are listed as follows :

1. Dimension 1 :

   Mean Red - The images are obtained in RGB form. As we know already, all colours can be made by varying the intensities of red,green and blue colours. In mathematical form, the image is broken into three independent planes, each plane containing the intensity of the corresponding colour in each pixel. In this manner, we obtain the red plane of the image with the help of openCV, and we extract the mean red intensity which is the sum of the red pixel intensities divided by the total number of pixels. Our first dimension is obtained as a result.

2. Dimension 2 :

   Standard Deviation Red - Following the method similar to obtaining dimension 1, we obtain the red plane of the image with the help of openCV, and we extract the STD, a quantity expressing by how much the red intensity of the pixels in the image differ from the mean value . Our second dimension is obtained as a result.

3. Dimension 3 :

   Mean Green - The images are obtained in RGB form. As we know already, all colours can be made by varying the intensities of red,green and blue colours. In mathematical form, the image is broken into three independent planes, each plane containing the intensity of the corresponding colour in each pixel. In this manner, we obtain the green plane of the image with the help of openCV, and we extract the mean green intensity which is the sum of the green pixel intensities divided by the total number of pixels. Our third dimension is obtained as a result.

4. Dimension 4 :

   Standard Deviation Green - Following the method similar to obtaining dimension 3, we obtain the green plane of the image with the help of openCV, and we extract the STD, a quantity expressing by how much the green intensity of the pixels in the image differ from the mean value . Our fourth dimension is obtained as a result.

5. Dimension 5 :

   Mean Blue - The images are obtained in RGB form. As we know already, all colours can be made by varying the intensities of red,green and blue colours. In mathematical form, the image is broken into three independent planes, each plane containing the intensity of the corresponding colour in each pixel. In this manner, we obtain the blue plane of the image with the help of openCV, and we extract the mean blue intensity which is the sum of the blue pixel intensities divided by the total number of pixels. Our fifth dimension is obtained as a result.

6. Dimension 6 :

   Standard Deviation Blue - Following the method similar to obtaining dimension 5, we obtain the blue plane of the image with the help of openCV, and we extract the STD, a quantity expressing by how much the blue intensity of the pixels in the image differ from the mean value . Our sixth dimension is obtained as a result.

7. Dimension 7 :

   Shannon's Entropy - a scalar value representing the entropy of an image I. Entropy is a statistical measure of randomness that can be used to characterize the texture of the input image. The implementation of our Shannon's entropy algorithm involves the usage of colour histograms, which is explained further in the following section.

   Color Histograms : Collect counts of color intensity values, organized in a set of bins.Since we know that the range of information value for this case is 256 values, we can segment our range in subparts (called bins) like:

$$[0, 255] = [0, 15] \cup [16, 31] \cup .... \cup [240, 255]$$
$$range = bin1 \cup bin2 \cup .... \cup binn = 15$$

   and we can keep count of the number of pixels that fall in the range of each bini.

Applying this to the example above we get the image below (axis x represents the bins and axis y the number of pixels in each of them).

Parameters of a color histogram dims: The number of parameters you want to collect data of. bins: It is the number of subdivisions in each dim. In our project, bins = 256 range: The limits for the values to be measured. In this case: range = [0,255] Figure 3.4 shows the blue histogram, figure 3.5 shows the green histogram and figure 3.6 shows the red histogram with the parameters of that color histogram dimensions.

After obtaining color histograms of each color (Red,Green and Blue), the values obtained in each bin are normalized. Then the frequency count for each colour is obtained and then the following formula is used to calculate the entropy value :

$$entropy+ = -(Hc/frequency(color)) * log10((Hc/frequency(color)))$$

Where Hc corresponds to the value at ith bin of the histogram.

In this manner, our seventh dimension is obtained.

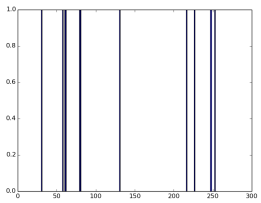


Figure 3.5: Blue Histogram

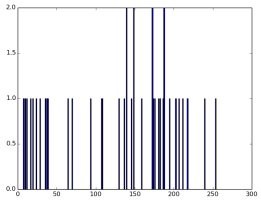


Figure 3.6: Green Histogram

8. Dimension 8 :

   White Contribution The rationale behind implementing this algorithm is to measure how much of the image is contributed by the color white which signifies the

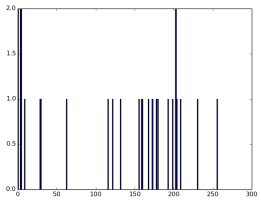


Figure 3.7: Red Histogram

foreground images. Black parts of the image are considered to be the background. Initially we are provided colour images. The first step involves converting the image into binary(black and white) form. In order to carry out this we use a technique in image processing known as thresholding which is explained below :

Thresholding of an Image : Separate out regions of an image corresponding to objects which we want to analyze. This separation is based on the variation of intensity between the object pixels and the background pixels. To differentiate the pixels we are interested in from the rest (which will eventually be rejected), we perform a comparison of each pixel intensity value with respect to a threshold (determined according to the problem to solve). In our case,its binary thresholding which is formulated as follows :

This thresholding operation can be expressed as:

$$dst(x,y) = \begin{cases} maxVal & \text{if } src(x,y) > thresh \\ 0 & \text{otherwise} \end{cases}$$

where src(x,y) is the intensity at that pixel value. If its higher than threshold value, it is marked as white(max value) else 0(black).

Following image thresholding, we obtain a new matrix of 0s and 255s and we calculate the percentage of 255s with respect to the total to get the white contribution of the image. Thus, our eigth dimension is obtained as a result.

9. Dimension 9 :

   Count Objects : Counting objects is one of the hardest techniques in image processing. It is a very abstract concept and there is no such documentation available which discusses a standardized algorithm to count objects. As a result, we use the

help of what algorithms are known to us and modify it in the manner explained further to obtain a count (not accurate due to plenty of noise, but its the difference in the values of these counts with other images which can help us determine something from the images). The key concepts behind implementing this algorithm are mentioned below.

2D Convolution : As in one-dimensional signals, images also can be filtered with various low-pass filters(LPF), high-pass filters(HPF) etc. LPF helps in removing noises, blurring the images etc. HPF filters helps in finding edges in the images. Essentially a convolution matrix is created and is applied to the image matrix. Based on the operation specified, the pixel under consideration takes up a value and a new convoluted image is obtained. If the convolution kernel is a LPF for the signal provided, it solves the purpose of removing substantial amount of noise. Figure 3.7 shows a reference image for the convolution step. Figure 3.8 shows the blur red image obtained from it.

Figure 3.8: Reference Image

Image Blurring (Image Smoothing) :

Image blurring is achieved by convolving the image with a low-pass filter kernel. It is useful for removing noises. It actually removes high frequency content (eg: noise, edges) from the image. So edges are blurred a little bit in this operation. Median Blur is used as the operation. In this,operation takes median of all the pixels under kernel area and central element is replaced with this median value. This is highly effective against salt-and-pepper noise in the images. Interesting thing is that, in the above filters, central element is a newly calculated value which may be a pixel value in the image or a new value. But in median blurring, central element is always replaced by some pixel value in the image. It reduces the noise effectively. Its kernel size should be a positive odd integer.

Thresholding : After obtaining the reduced noise version of the image, binary

Figure 3.9: Blur Red Image

thresholding is carried out similar to what was explained previously. After this we proceed to the contour finding stage. Figure 3.9 shows the result of thresholding of the reference image.

Figure 3.10: Thresholding of the Image

Contours : Contours can be explained simply as a curve joining all the continuous points (along the boundary), having same color or intensity. The contours are a useful tool for shape analysis and object detection and recognition. For better accuracy,we use thresholding to obtain binary images. Finally the number of such contours are obtained which is returned to obtain our final dimension.

3.3 *Spline Regression*

A spline curve is a mathematical representation which allow a user to design and control the shape of complex curves and surfaces. The user enters a sequence of points, and a curve is constructed whose shape closely follows this sequence.The points are called control points. A curve that actually passes through each control point is called an interpolating curve and a curve that passes near to the control points but not necessarily through them is called an approximating curve. Figure 3.10 shows the algorithm for the semi supervised feature selection via spline regression.

Cubic Splines are used for the following reasons

1. it is the lowest degree polynomial that can support an inflection

Algorithm 1 Semisupervised Feature Selection via Spline Regression (S^2FS^2R)

Input: matrix of n training videos $X = [x_1, \ldots, x_n] \in \mathbb{R}^{d\times n}$, $X^{\mathcal{L}} = [x_1, \ldots, x_{n_l}] \in \mathbb{R}^{d\times n_l}$ is a matrix of first $n_l (n_l \leq n)$ labeled video samples and $Y^{\mathcal{L}} = [y_1, \ldots, y_{n_l}] \in \{0, 1\}^{c\times n_l}$ is the corresponding indicator matrix for c labels (or semantic categories); $X^{\mathcal{U}} = [x_{n_l+1}, \ldots, x_{n_l+n_u}] \in \mathbb{R}^{d\times n_u}$ is a matrix of unlabeled videos whose labels are not given; k is the number of the nearest neighbors in local clique $\mathcal{N}_i$ for each video x_i; control parameter μ and regularization parameter λ; f is the number of features to be selected.

Output: index idx of the top f selected features

1: **for** each video $x_i \in X$ **do**
2: Construct local clique $\mathcal{N}_i$ by adding x_i with its $k-1$ nearest neighbors;
3: Construct matrix $\mathbf{K}_i$ using Green's function $G_{i,j}$ defined on $\mathcal{N}_i$;
4: Construct matrix $A = \begin{pmatrix} \mathbf{K}_i & P \\ P^T & \mathbf{0} \end{pmatrix}$;
5: Construct matrix M_i which is the up left $k \times k$ submatrix of the matrix A^{-1};
6: **end for**
7: Form matrix $\mathcal{D}$ using Eq. (5);
8: Form matrix $\mathcal{A}$ using Eq. (4);
9: Form matrix $\mathcal{M}$;
10: Set $t = 0$ and initialize $D_{(0)} \in \mathbb{R}^{d\times d}$ to be an identity matrix;
11: **repeat**
12: $U_{(t)} = \mathcal{M} + \lambda D_{(t)}$;
13: $W_{(t)} = [u_1, \ldots, u_c]$ where $u_1, \ldots, u_c$ are the eigenvectors of $U_{(t)}$ corresponding to the first c smallest eigenvalues;
14: Update matrix $D_{(t+1)}$ as
$$D_{(t+1)} = \begin{bmatrix} \frac{1}{2||w^1_{(t)}||_2} & & \\ & \cdots & \\ & & \frac{1}{2||w^d_{(t)}||_2} \end{bmatrix};$$
15: $t = t + 1$;
16: **until** convergence.
17: Sort each feature of the jth video sample $X_{(j,i)}|_{i=1}^{d}$ according to the value of $||w_i||_2$ in descending order;
18: Output the index idx of the top f selected features.

Figure 3.11: Semi Supervised Feature Selection Via Spline Regression

2. it is very well behaved numerically that means that the curves will usually be smooth and not jumpy

To make more complex curves we used the concepts of piecewise curve fitting and parameterization. The conditions for piece wise curve fitting are as follows:

1. We require that each curve segment pass through its control points. Thus,

$$f(xi) = yi, \tag{1}$$

and,

$$fi(xi + 1) = yi + 1. \tag{2}$$

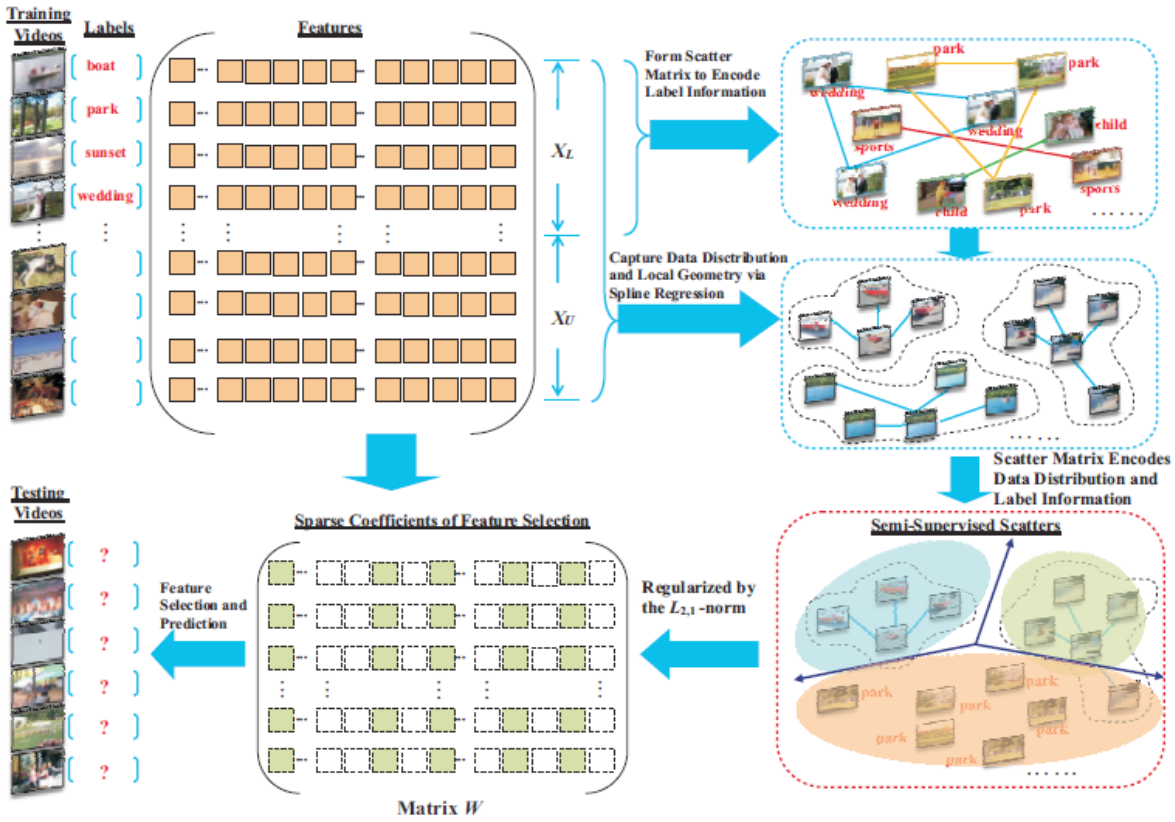


Figure 3.12: Supervised Feature Selection Via Spline Regression

This enforces C0 continuity that is,where the curves join they meet each other.

2. We require that the curve segments have the same slope where they join together. Thus,

$$f'(i)(xi+1) = f'(i+1)(xi+1). \tag{3}$$

This enforces C1 continuity that is that slopes match where the curves join.

3. The curve segments have the same curvature where they join together. Thus,

$$f''(i)(xi+1) = f''(i+1)(xi+1). \tag{4}$$

This enforces C2 continuity that curvatures match at the join.

Semi Supervised Feature Selection using Spline Regression Methodology. Figure 3.11 shows the methodology for semi supervised feature selection via spline regression.

Given a large data set of labelled and unlabelled videos, where the size of unlabelled is greater than the size of labelled, we employ semi supervised feature selection using spline regression in the following manner :

1. We create a feature matrix with the entire data set.

2. We create a scatter matrix with only the labelled set to encode label information

3. With the unlabelled set,we apply spline regression to capture data distribution and local geometry

4. We create a scatter matrix with the same and we end up with semi supervised scatters in the process

5. After regularization via the L21 norm, the new matrix W will be sparse in rows which makes it ideal for feature selection

6. The top features are taken into consideration and is used to classify the test videos

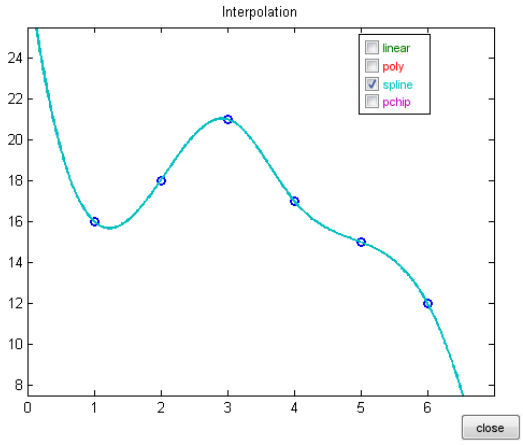


Figure 3.13: Supervised Feature Selection Via Spline Regression Interpolation

Some of the challenges that spline regression faces is that there is a tendency to overfit and how these control points are determined. Fortunately, many programming language implementations of this automatically assign these control points in such a way that piecewise smooth curves are generated which follow the aforementioned three conditions. Figure 3.12 shows the spline regression interpolation using supervised feature selection. The resultant curve generated out of this process is made to best fit the data without overfitting the data.

3.4 *Graph Based Semi Supervised Algorithm*

An approach for the semi supervised learning is formulated that is based on Gaussian random field model. All the labeled and the unlabeled nodes (data) are represented as vertices in a weighted graph, and the edges encode the similarity between the nodes connecting them. The learning problem is then proposed in terms of a random Gaussian field on the graph, where the mean of the field is basically characterized in harmonic functions, and it is efficiently obtained using the matrix operations and the associated methods. The learning algorithms resulting from this have intimate connections with random walks, spectral graph theory and electric networks. Methods has been incorporated to calculate the harmonic function and also a method of parameter learning by entropy minimization. Experimental results for applying semi supervised learning on the video datasets has been observed.

Basic framework

Suppose there are l labeled points (x1,y1),(x2,y2),(x3,y3),...,(xn,yn) and u unlabeled points x(l+1),x(l+2),x(l+3),...,x(l+u); such that $l << u$.

Let the total number of instances be n such that n = l + u. Consider a connected graph G = (V,E) with nodes V corresponding to the n data points, with nodes L = 1,2,3,...,l corresponding to the labeled points with labels y1, y2, y3, ..., yl, and nodes U = l+1,l+2,l+3,...,l+u corresponding to the unlabeled points. The main task is to assign labels to nodes U. Assume there is an n * n symmetric weight matrix W on the edges of the graph. For example, when $x \in R^m$, the weight matrix can be

$$w(ij) = e^{-\frac{\sum((x(id)-x(jd))^2)}{\sigma^2}}$$

where d = 1...m and $\sigma 1$,$\sigma 2$,$\sigma 3$, ... , σm are length scale hyperparameters for each dimension. Thus it is evident that the nearby points in Euclidean space are assigned large edge weight.

The strategy is to compute a real-valued function $f : V- > R$ on G with certain very nice properties, and then assign labels based on f. We constrain f to take values from the unique class points that must have been assigned to one the label and not otherwise. This motivates the choice of the quadratic energy function

$$E(f) = \frac{\sum W(ij)(f(i) - f(j))^2}{2}$$

The harmonic property means that the value of f at each unlabeled data point is the average of f at neighbouring points:

$$f(j) = \frac{\sum w(ij)f(i)}{d(j)}$$

for j = l+1, l+2, l+3, ... , l+u which is consistent with prior smoothness of f with respect to the graph. Let the W be the weight matrix.

$$W = \begin{vmatrix} W_{ll} & W_{lu} \\ W_{ul} & W_{uu} \end{vmatrix}$$

Letting $f = \begin{vmatrix} f_l \\ f_u \end{vmatrix}$ where f_u denotes the values on the unlabeled data points, the harmonic solution $\triangle f = 0$ subject to $f|_L = f_l$ is given by

$$f_u = (D_{uu} - W_{uu})^{-1} W_{ul} f_l$$

The algorithm focus on the above harmonic function as a basis for semi supervised classification. However, the Gaussian random field model from which this function is derived basically provides the learning framework such that it has a consistent probabilistic semantics.

$L = D - W$ where L is the Laplacian Matrix, D is the diagonal matrix such the each of the diagonal element gives the $\sum W_{ij}$ where $j = 1, 2, 3, ..., n$. L represents the $\triangle$.

It is an approach to semi-supervised learning based on a Gaussian random field model defined with respect to a graph which is weighted and representing labeled data and unlabeled data. Experimental results for video classification, demonstrating that the framework has the potential to effectively utilize the structure of unlabeled data to improve classification accuracy.

The graph is assumed to be totally connected. Therefore, The overall complexity of the graph based semi-supervised algorithm is $O(n^2 * d)$ where n is the number of nodes in the graph and d is the number of classes.

3.5 *AdaBoost Algorithm*

Boosting is a general method for improving the accuracy of any given learning algorithm. Boosting refers to a general and provably effective method of producing a very accurate

prediction rule by combining rough and moderately inaccurate rules of thumb.

The AdaBoost algorithm as introduced by Schapire and Freund, is able solve many of the problems of the earlier boosting algorithms. Pseudocode for AdaBoost is given in figure. The algorithm takes as input a training set $(x1, y1), ..., (xm, ym)$ where each xi belongs to some domain or instance space X , and each label y_i is in some label set Y . For the time being, we assume Y = $\{-1, +1\}$. AdaBoost calls a given weak or base learning algorithm repeatedly in a series of rounds $t = 1, ..., T$. One of the main ideas of the algorithm is to maintain a distribution or set of weights over the training set. The weight of this distribution on training example i on round t is denoted by $D_t(i)$. Initially, all weights are set equally, but on each round, the weights of incorrectly classified examples are increased so that the weak learner is forced to focus on the hard examples in the training set.

The weak learners job is to find a weak hypothesis h(t) : X $\rightarrow \{-1, +1\}$ appropriate for the distribution D(t) . The goodness of a weak hypothesis is measured by its error

Notice that the error is measured with respect to the distribution D(t) on which the weak learner was trained. In practice, the weak learner may be an algorithm that can use the weights D(t) on the training examples. Alternatively, when this is not possible, a subset of the training examples can be sampled according to D(t) , and these (unweighted) resampled examples can be used to train the weak learner.

Once the weak hypothesis h(t) has been received, AdaBoost chooses a parameter a(t) as in the figure. Intuitively, a(t) measures the importance that is assigned to h(t). Note that $a(t) >= 0$ if $E(t) <= 1/2$ (which we can assume without loss of generality), and that a(t) gets larger as E(t) gets smaller. Figure 3.13 shows the algorithm for AdaBoost algorithm for multiclass classification.

The distribution D(t) is next updated using the rule shown in the figure. The effect of this rule is to increase the weight of examples misclassified by h(t) , and to decrease the weight of correctly classified examples. Thus, the weight tends to concentrate on hard examples.

The final hypothesis H is a weighted majority vote of the T weak hypotheses where a(t) is the weight assigned to h(t).

Multiclass classification

So far, we have only considered binary classification problems in which the goal is to

Given: $(x_1, y_1), \ldots, (x_m, y_m)$ where $x_i \in X$, $y_i \in Y = \{-1, +1\}$
Initialize $D_1(i) = 1/m$.
For $t = 1, \ldots, T$:

- Train weak learner using distribution D_t.
- Get weak hypothesis $h_t : X \to \{-1, +1\}$ with error

$$\epsilon_t = \Pr_{i \sim D_t} \left[h_t(x_i) \neq y_i \right].$$

- Choose $\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right)$.
- Update:

$$\begin{aligned} D_{t+1}(i) &= \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases} \\ &= \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t} \end{aligned}$$

where Z_t is a normalization factor (chosen so that D_{t+1} will be a distribution).

Output the final hypothesis:

$$H(x) = \operatorname{sign} \left(\sum_{t=1}^{T} \alpha_t h_t(x) \right).$$

Figure 3.14: AdaBoost Algorithm

distinguish between only two possible classes. Many (perhaps most) real-world learning problems, however, are multiclass with more than two possible classes. There are several methods of extending AdaBoost to the multiclass case.

The most straightforward generalization , called AdaBoost.M1, is adequate when the weak learner is strong enough to achieve reasonably high accuracy, even on the hard distributions created by AdaBoost. However, this method fails if the weak learner cannot achieve at least 50 percent accuracy when run on these hard distributions.

For the latter case, several more sophisticated methods have been developed. These generally work by reducing the multiclass problem to a larger binary problem. Schapire and Singers algorithm AdaBoost.MH works by creating a set of binary problems, for each example x and each possible label y , of the form: For example x , is the correct label y or is it one of the other labels? Freund and Schapires algorithm AdaBoost.M2 (which is a special case of Schapire and Singers AdaBoost.MR algorithm) instead creates binary problems, for each example x with correct label y and each incorrect label y' of the form: For example x , is the correct label y or y' ?

These methods require additional effort in the design of the weak learning algorithm. A different technique, which incorporates Dietterich and Bakiris method of error-correcting output codes, achieves similar provable bounds to those of AdaBoost.MH and AdaBoost.M2, but can be used with any weak learner which can handle simple, binary labeled data.

Schapire and Singer give yet another method of combining boosting with error-correcting output codes.

Advantages Of AdaBoost Algorithm

1. It is fast, simple and easy to program.
2. It has no parameters to tune (except for the number of round T).
3. It requires no prior knowledge about the weak learner and so can be flexibly combined with any method for finding weak hypotheses.
4. Finally, it comes with a set of theoretical guarantees given sufficient data and a weak learner that can reliably provide only moderately accurate weak hypotheses. This is a shift in mind set for the learning-system designer: instead of trying to design a learning algorithm that is accurate over the entire space, we can instead focus on finding weak learning algorithms that only need to be better than random.

3.6 *Performance Analysis :*

Python provides a set of modules under the MatPlotLib library which enables us to construct all sorts of graphs and charts which can be used to pictorially depict the performance of the algorithm as well as the speed of our computation. Accuracy with respect to our training and test sets are compared and we also check the type of relationship exhibited by computational time vs accuracy.

After training our algorithm, the initially unsupervised videos, will now have labels assigned to them (with an acceptable level of error) and now we can pass our test videos against this improved training set to semantically recognize them.

We carry out these procedures to perform a comparative study on whether modifying certain parameters based on the data set under consideration will have an effect in increasing/decreasing computational time and accuracy. Based on the results obtained after this step, we determine what the best conditions are for our implementation.

4 Work Done and Results

At the end of our implementation thus far, we have been able to create a feature matrix corresponding to each video example with the help of Open Cv's Image processing tools. The feature vector has values which are common to all images so far so it can be seen that all videos are so far considered to be on a level playing field. The disadvantage as mentioned previously is the time it takes to compute these features and the further problem of using these features in the learning algorithms. It certainly calls for the need of feature reduction. In addition to this, the graph algorithm for semi supervised learning is implemented on the Video datasets and experimental results for the same has been noted. There are three classes of videos that has been considered for now. Human, nature and walk.

Advantages

1. Adopting Gaussian Fields over a continuous state space rather than random elds over the discrete label set.

2. This relaxation to a continuous rather than discrete sample space results in many attractive properties.

3. The most probable conguration of the Field is unique, is characterized in terms of harmonic functions, and has a closed form solution that can be computed using matrix methods.

4. The learning methods introduced here have intimate connections with random walks, electric networks, and spectral graph theory, in particular heat kernels and normalized cuts.

Disadvantages

1. The implementation is only based on the mean of the Field, which is characterized in terms of harmonic functions and spectral graph theory. A more comprehensive approach will be to use the probabilistic semantics approach to improve the classification approach can be proposed.

2. Instances are assumed to have a totally connected graph. Therefore, computing the weight matrix W takes the complexity of $n * n * d$ where d is 9. Sparsifying the

graph can help a lot here. Figure 4.1 shows the dimensions extracted for the first six frames for Video 0. Figure 4.2 shows the feature matrix for one frame. Figure 4.3 shows the feature matrix for one video.

Video No.	Frame no.	Mean Red	Std Red	Mean Blue	Std Blue	Mean Green	Std Green	Entropy	White Contrib	Count objects
0	0	1.32E+002	1.24E+001	1.28E+002	1.11E+001	1.33E+002	1.24E+00	4.90E+000	3.51E-001	4.00E+000
	1	1.30E+002	2.75E+001	1.25E+002	2.67E+001	1.31E+002	2.75E+001	5.37E+000	6.67E-002	4.00E+000
	2	1.33E+002	2.74E+001	1.28E+002	2.64E+001	1.34E+002	2.74E+001	5.38E+000	7.06E-002	6.00E+000
	3	1.36E+002	1.10E+001	1.31E+002	9.85E+000	1.37E+002	1.11E+001	4.70E+000	3.60E-001	4.00E+000
	4	1.34E+002	1.04E+001	1.29E+002	9.48E+000	1.35E+002	1.05E+001	4.67E+000	3.76E-001	6.00E+000

Figure 4.1: Dimensions for first 6 frames for Video 0

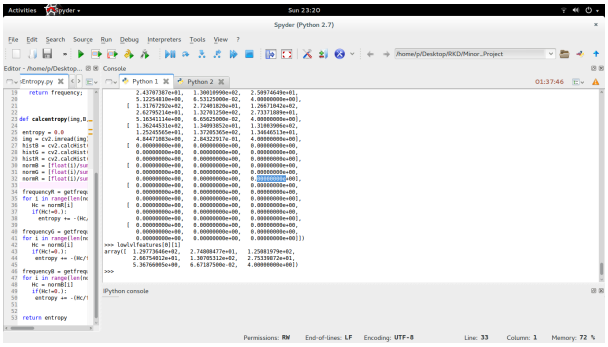


Figure 4.2: Feature Matrix For One Frame

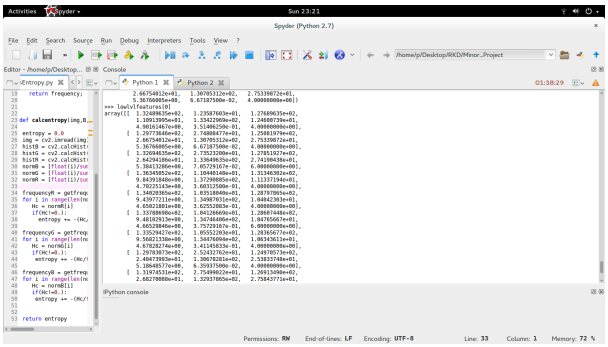


Figure 4.3: Feature Matrix For One Video

Predicting Unlabeled Instances :

In figure 4.4, Results shows a matrix of $(n-l)*C$ where n is the number of video instances, including both the labeled and the unlabeled instances and l is the number of labeled instances and C is the number of classes to which the videos can be classified. $n = 34, l = 9, C = 3$.

The matrix f_u represents the 1 for the maximum value obtained for the unlabeled instances for the real valued function $f : V \rightarrow R$ and 0 for the other columns with 1

representing that the unlabeled instance is of that particular class and 0 represents it is not of that class where V represents the set of vertices aka instances.
The three columns represents the real values for unlabeled instances to which class they belong to. More the score, more the probability that the video instance will belong to that particular class.

As compared with the unlabeled instances, the matrix has shown promising results and using scores to predict about the type of dataset it refers, the accuracy is about 75%. Considering that the number of labeled videos is small, the accuracy of about 75% indicates that the graph algorithm is showing promising results and with more input labeled datasets and unlabeled datasets, the accuracy is expected to increase.

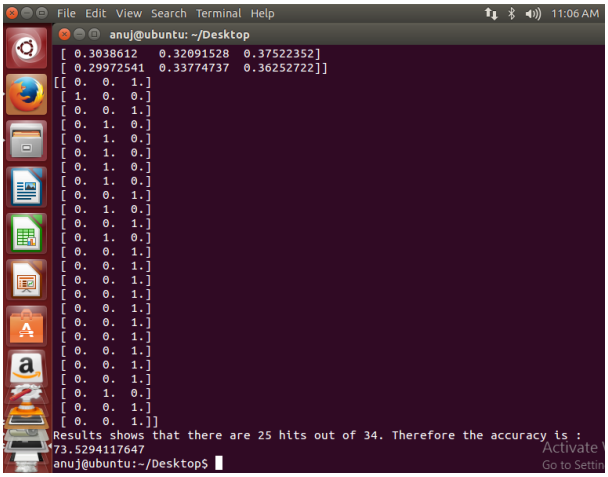


Figure 4.4: Predicted Unlabeled Instances

5 Conclusion and Future Work

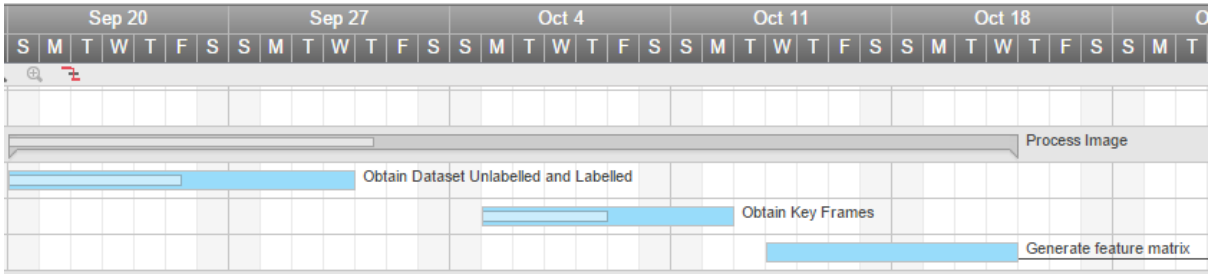
To sum up the progress thus far, Keyframe and feature extraction has been completed along with the graph learning algorithm which uses gaussian fields and the harmonic functions to predict the classes of the unlabeled nodes using the labeled nodes and unlabeled nodes and the labeled classes. The predicted function is normalized over the set such that the set contains all the labeled classes atleast once. And thus it uses the semi supervised learning

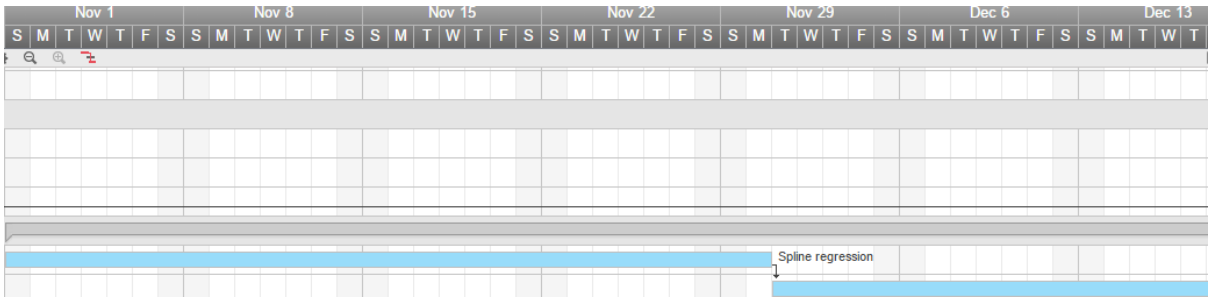
There is an urgent need to reduce the features in order to speed up computation when it comes to training the videos and testing it as well. Spline Regression is the logical next step required in order to reduce these features. It helps preserving local geometry and data distribution and is a very good technique to implement for semi supervised learning. Once this has been carried out, we feed this normalized,reduced feature matrix through the graph algorithm and thereby complete the learning. Another algorithm yet to implement is the Adaboost technique for multi class classification. The test set then needs to be provided to check the validity of our implementation and these results are to be compared and contrasted.

Clearly, there is a lot of scope and improvement in the methods but there is a great deal of promise in what the results hold.

6 Time Line Of The Project

Process Image	**09/20/15**	**10/21/15**		**24d**
Obtain Dataset Unlabelled and Labelled	09/20/15	09/30/15	Anuj	9d
Obtain Key Frames	10/05/15	10/12/15	Siddharth R	6d
Generate feature matrix	10/14/15	10/21/15	Rohit	6d
Apply learning algorithms	**11/01/15**	**01/29/16**		**66d**
Spline regression	11/01/15	11/30/15	Anuj	22d
Graph Method	12/01/15	12/30/15	Siddharth R	22d
Adaboost	12/31/15	01/29/16	Rohit	22d
Compare Results	**02/01/16**	**03/17/16**		**34d**
Plot Graphs	02/01/16	02/09/16	Anuj	7d
Compare the results	02/11/16	02/29/16	Siddharth R	13d
Note Results	03/01/16	03/17/16	Rohit	13d





7 References

References

[1] R. Ewerth and B. Freisleben
Semi-supervised learning for semantic video retrieval in *Proceedings of the 6th ACM International Conference on Image and Video Retrieval.*
2007, pp. 154 161.

[2] Yahong Han, Yi Yang, Yan Yan, Zhigang Ma, Nicu Sebe, Xiaofang Zhou
Semisupervised Feature Selection via Spline Regression for Video Semantic Recognition
IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, VOL. 26, NO. 2, FEBRUARY 2015

[3] Andrew Goldberg and Xiaojin Zhu
Seeing stars when there aren't many stars: Graph-based semi-supervised learning for sentiment categorization.
In *HLT-NAACL 2006 Workshop on Textgraphs: Graph-based Algorithms for Natural Language Processing, New York, NY, 2006.*

[4] Xiaojin Zhu, John Lafferty, and Zoubin Ghahramani
Semi-supervised learning: From Gaussian fields to Gaussian processes.
Technical Report CMU-CS-03-175, Carnegie Mellon University, 2003.

[5] S. Dagtas, W. Al-Khatib, A. Ghafoor, and R. L. Kashyap
Models for motion-based video indexing and retrieval
IEEE Transactions on Image Processing
pp. 88 101, 2000.

[6] Y. Yang, H. Shen, Z. Ma, Z. Huang, and X. Zhou
2,1-norm regularized discriminative feature selection for unsupervised learning
Twenty-Second International Joint Conference on Artical Intelligence (IJCAI-11) 2011, pp. 1589 1594.

[7] Z. Xiaojin

Semi-supervised learning literature survey

Computer Science, University Wisconsin-Madison, Madison, WI, Technical Report vol. 1530, 2007.

[8] Bryan R. Gibson, Timothy T. Rogers, and Xiaojin Zhu.

Human semi-supervised learning. *Topics in Cognitive Science pp. 132-172, 2013*

[9] Xiaojin Zhu and Andrew B. Goldberg

Introduction to Semi-Supervised Learning.

Morgan and Claypool, 2009.

[10] Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty.

Semi-supervised learning using Gaussian fields and harmonic functions.

In The 20th International Conference on Machine Learning (ICML), 2003