DISTANCE METRIC LEARNING FOR DATA CLASSIFICATION

SUMMER INTERNSHIP REPORT 10th MAY TO 5th JULY 2018

Megh Manoj Bhalerao
Department of Electrical & Electronics Engineering
NITK-Surathkal
megh.bhalerao@ieee.org

Priyadarshini Kumari
Department of Electrical Engineering
IIT-Bombay
priyadarshini.kri15@gmail.com

Prof. Subhasis Chaudhuri
Department of Electrical Engineering
IIT-Bombay
sc@ee.iitb.ac.in

August 18, 2018

Contents

1 Abstract

Machine Learning algorithms such as K-Means Clustering, K-Nearest Neighbors, Support Vector Machines(SVMs) *etc.* rely on a good metric to be specified over the input data. The standard euclidean norm between data-points does not always give us a good measure of the relation between data points. Hence it is crucial to specify a distance metric over the input data. This process of estimating a distance metric by utilizing information inherent to the data is called Distance Metric Learning(DML). Former approaches to DML make the use of class information to learn the distance metric over the dataset. Unlike the previous approaches, our approach makes the use of relative information of similarity or dissimilarity of the data-points rather than defining strict classes to separate the data-points. We obtain the information of similarity and dissimilarity of the data-points by using human perception of the same and attempt to learn a distance metric so as to mimic this human response. Initially, we attempt to apply DML upon a haptic dataset containing acceleration response signals (after being perceived by the sense of touch) from various materials with the help of a confusion matrix which gives us the human perception of similarity and dissimilarity. We then project our data-points into a new vector space defined by our learned distance metric to minimize the distance between similar pairs and maximize the distance between the dissimilar pairs.

2 Introduction & Scope

Distance Metric Learning is generally chosen as a preprocessing step to many machine learning algorithms. Xing *et al.*[1] showed that the accuracy of K-means and constrained K-Means is brought to 100% after a distance metric is learned over the input data and the data-points are projected into a new vector space based on the learned metric. This aforementioned paper was the beginning of a solid research in DML. Here, DML is applied to another dataset as well, which is the Braille Character Dataset [2] by Jack M. Loomis. The requirements for a distance metric to be learned is a confusion matrix (also called a similarity matrix) and feature vectors corresponding to the data-points (obtained using some feature extraction techniques from the raw data).

The entire problem can be discretized into the following subsections:

1. Develop a DML algorithm to cluster the haptic signals into similar and dissimilar clusters and evaluate its accuracy on a testing set.

2. Apply a least squares regression DML algorithm for learning a distance metric on the Braille Character Dataset using different feature vectors corresponding to various Peano curves.

3. Estimate a distance matrix corresponding to the Braille Character Dataset (using the feature vector and the learned distance metric), *i.e.* the pairwise distances between the feature vectors of each of the alphabets. Compare the estimated distance matrix with the human response distance matrix.

As of now, a relaxed version of 1 is completed. 2 and 3 are completed. Furthermore, in the subsequent sections, 1 and 2 describe about the haptic and braille character dataset respectively.

3 The Datasets

3.1 The Haptic Dataset

3.1.1 The Acceleration Responses

690 Haptic acceleration responses collected from a total of 69 materials (10 signals from each sample) are obtained from the TUM repository[3]. One sample is taken out randomly from the 10 samples of each of the material and used for the DML algorithm developed here. These samples are preprocessed as follows:

1. These 3-D signals need to be converted into 1-D. This conversion is achieved using the 3D DFT[4] function.
2. The dimension of each of the data sample is 42000 which needs reduction since handling high dimensions is computationally expensive.
3. The dimensionality of the data is reduced to 6 dimensions using the technique of feature extraction by a constant Q factor filter bank of a Gaussian window. For details of why such a filter bank is chosen, the work of Sliman *et al.*[7] can be referred to.
4. Hence now we have our data (say X) in the form of a 69×6 matrix with each row giving us the frequency response of every data-point.

3.1.2 The Confusion (Similarity) Matrix

1. This is a 69×69 matrix also taken from the TUM repository (say C).
2. The $(i, j)^{th}$ entry of this confusion matrix tells us the number of people who thought that the i^{th} and the j^{th} substance to be the same. The experiment to acquire the confusion matrix was carried out with a total of 30 people.

3.1.3 Triplets and Slack Variables

1. Considering the X and C matrices, we now generate triplets of the form (i, j, k) which encapsulate the following meaning : According to human perception, the substance x_i is more similar to x_j than it is to x_k. Also define $D = 30 - C$.
2. The triplets are randomly generated by the following algorithm:

Algorithm 1 : Triplet Generation Algorithm

Specify the number of random triplets to be generated say n. Also define a matrix $T[n \times 3]$ which stores all the randomly generated triplets
while $i \neq n$ **do**
  Generate 3 random numbers between 1 and 69 and let them be (i, j, k)
  if $D(i, j) < D(i, k)$ **then**
    Add the triplet (i, j, k) to T
    $i \leftarrow i + 1$
  end if
end while

3. The slack variables account for any unsatisfied constrains. The slack variables(ε_{ijk}) are calculated as follows:

Algorithm 2 : Calculation of Slack Variables

1: Define a matrix $Q = D/30$ which can be thought of as the normalized version of the D matrix .We divide by 30 since that is the total number of people with whom the experiment is carried out. Also define a variable ϵ to store the values of all ε_{ijk}
2: **for** all $(i, j, k) \in T$ **do**
3: $\quad \varepsilon_{ijk} = Q(i, k) - Q(i, j)$
4: **end for**

3.2 The Braille Character Dataset

3.2.1 The Confusion (Similarity) Matrix

This data is obtained from Jack M.Loomis's paper(1985). This confusion matrix is a 26×26 matrix where each of the row/column corresponds to a letter of the English alphabet. Let this matrix be called C_b. The entries of this matrix are interpreted as follows :

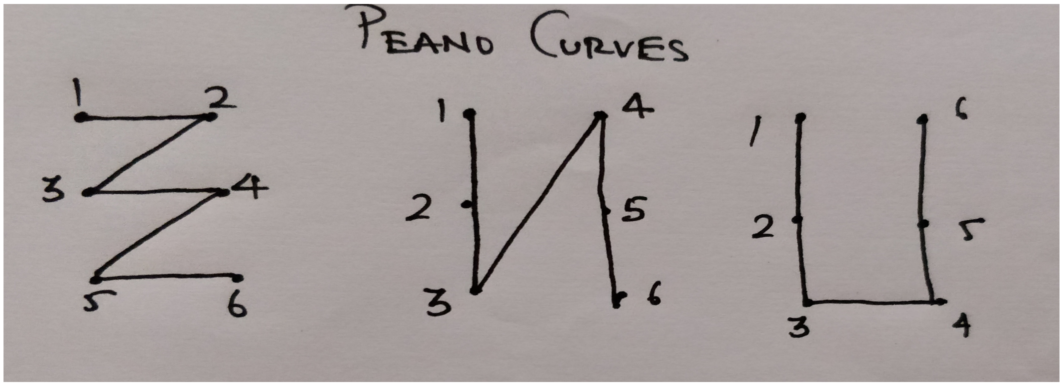
1. Consider any i^{th} row and j^{th} column of C_b *i.e.* $(i, j)^{th}$ entry of C_b.
2. $C_b(i, j)$ indicates the fraction of people who responded with the j^{th} alphabet when the stimulus corresponding to the i^{th} alphabet was provided.
3. This tells us that the matrix need not be a symmetric matrix.
4. The matrix is converted to the standard form of a confusion matrix by symmetrizing it.

3.2.2 The Chosen Feature Vectors

The feature vectors are chosen by following various configurations of Peano Curves. Peano Curves are defined as continuous lines between a set of points, starting from a constant chosen point and do not cross each other. Here, Peano Curves are be drawn for the braille characters since they are nothing but a set of 6 points of which some are raised and the others aren't. The feature vectors are chosen according to the following procedure:

1. Peano Curves are drawn with the help of 6 dots of the braille characters.
2. The order in which the dots are connected decide the numbering of the dots in the braille alphabet.
3. The dots of the braille alphabet are numbered accordingly.
4. The feature vector for each alphabet is a six dimensional vector with 1's at the places where the braille dots are raised and 0 otherwise.
5. Hence, the complete feature vector (say X_b) is a 26×6 vector with each row corresponding to the feature vector of one alphabet.

The feature vectors corresponding to Peano Curves of Row Stacking, Column Stacking and U Shaped Stacking are taken for experimentation and shown below respectively.



4 Problem Formulation

4.1 For the Haptic Dataset

Our goal is to use DML to learn a distance metric M over the dataset X using the confusion matrix C. After learning the distance metric, we project all our data points into a new vector space by the projection matrix L where $L^T L = M$. The process of learning this distance metric M is achieved by the following convex optimization problem :

$$\begin{aligned} \underset{M}{\operatorname{argmin}} \quad & \operatorname{trace}(M) + \sum_{ijk} \varepsilon_{ijk} \\ \text{subject to} \quad & (Ar, M) < 1 - \varepsilon_{ijk} \; \forall \, (i,j,k) \in T, \\ & M \succeq 0. \end{aligned}$$

where

$$Ar = (x_i - x_j)(x_i - x_j)^T - (x_i - x_k)(x_i - x_k)^T$$

and x_i, x_j & x_k are the i^{th}, j^{th} & k^{th} row of the matrix X.

4.2 For the Braille Character Dataset

The metric learning algorithm for the Braille Dataset is formulated as a least squares regression problem of the form :

$$\underset{M}{\operatorname{argmin}} \, \|P^{\mathsf{T}} M P - D_{b_{jk}}\|$$

subject to

$$M \succeq 0$$

The terms are defined as follows:

1. P : The vector containing the difference between all the possible pairs of the feature vectors of the Braille Character Set.
2. M : The Distance Metric that is to be learnt.
3. $D_{b_{jk}}$: The $(j,k)^{th}$ element of the human response distance matrix.

5 Algorithm

5.1 For the Haptic Dataset

1. The convex optimization algorithm is tackled using the method of iterative projections into the two constrain spaces until the constrains are satisfied for all the triplets. We say that our learned distance metric M converges when the constrains are satisfied for all the triplets. The implementation of the algorithm was done using MATLAB R2017a.

Algorithm 3 : Convex Optimization Algorithm - An example of a semi-definite program

---

Generate a certain number of random triplets of the form (i, j, k), using Algorithm 1, which encode the following information: the data-point x_i is closer to the data-point x_j than it is to x_k.
Let it be the number of iterations that have happened up-to a certain point. Choose a suitable learning rate α.
Generate a random guess initially for the distance matrix M. Force it to be PSD by imposing the standard eigenvalue condition. Initialize a variable $s = 0$ and $\varepsilon_{sum} = 0$
for each triplet (i, j, k) in T **do**
    Calculate the Ar matrix.
    $s = s + Ar$ and $\varepsilon_{sum} = \varepsilon_{sum} + \varepsilon_{ijk}$
    while $(s, M) < it - \varepsilon_{sum}$ **do**
        $M \leftarrow M + (it - (s, M)) \times \frac{\nabla(s,M)}{\|s\|}$
    end while
    Force $M \succeq 0$
    $M \leftarrow M + \alpha \times I$
end for

---

2. Another approach was attempted using the standard CVX (Convex Optimization) package for MATLAB. This package is authored by Stephen P. Boyd *et al.*[5]. The constrains that are imposed in this approach are more relaxed compared to the original formulation. The original problem imposes a margin upon the difference between the similar and dissimilar pair of data-points. The relaxed version just requires that the distance between the similar pair be lesser than the distance between the dissimilar pair. The code snippet of problem formulation using CVX is given below.

```
cvx_begin
variable M(6,6) symmetric semidefinite
minimize (trace(M))
subject to
```

```
Ars*(unroll(M)) >= 0;
cvx_end
t = (Ars*(unroll(M)) >= -10^(-nErr));
```

Brief Explanation of the above code snippet:

(a) M : The distance metric M which is initially itself defined to be a Symmetric and a Positive Semidefinite Matrix.

(b) Ars : It is the matrix whose every row is the transpose of the column unrolled matrix of each of the Ar matrices as defined previously.

(c) nErr : It is the numerical approximation to which each of the constrain is satisfied.

(d) t : It is a column vector indicating how many of the constrains have been satisfied for a given value of numerical error. 1 indicates that the constrain is satisfied and a 0 indicates that it isn't.

5.2 For the Braille Character Dataset

As mentioned above the optimization problem here is a least squares regression algorithm and it has a closed form solution which is implemented using the algorithm given below:

Algorithm 4 : Learning M using least squares regression - closed form solution

---

1: Input the feature vector(X_b) and the confusion matrix(C_b). Let the i^{th} and j^{th} entries of X be x_{b_i} & x_{b_j} respectively.
2: Define a Distance Matrix $D_b = 1 - C_b$. Let the $(i,j)^{th}$ entry of D_b be $d_{b_{ij}}$
3: Compute the pairwise distance between all the possible pairs of feature vector i.e. $x_{b_i} - x_{b_j}$ and store it in a separate variable say P.
4: Our optimization problem is a least squares regression problem of the form

$$\underset{M}{\operatorname{argmin}} \|P^{\mathsf{T}} M P - D\|$$

5: The problem is brought in its standard form by writing it in a slightly different manner i.e.

$$\underset{M}{\operatorname{argmin}} \|Hm - D\|$$

where m is the unrolled vector of the matrix M.
6: Let the i^{th} entry of P be p_i. The i^{th} row of H is defined as the unrolled row vector of the matrix $p_i^{\mathsf{T}} \times p_i$.
7: The closed form solution of the above regression problem is given by

$$M = (H^{\mathsf{T}} H)^{-1} H^{\mathsf{T}} E$$

where E is the unrolled column vector of the lower triangular part of the matrix D.

---

6 Accuracy Check

6.1 For the Haptic Dataset when the constrains are relaxed

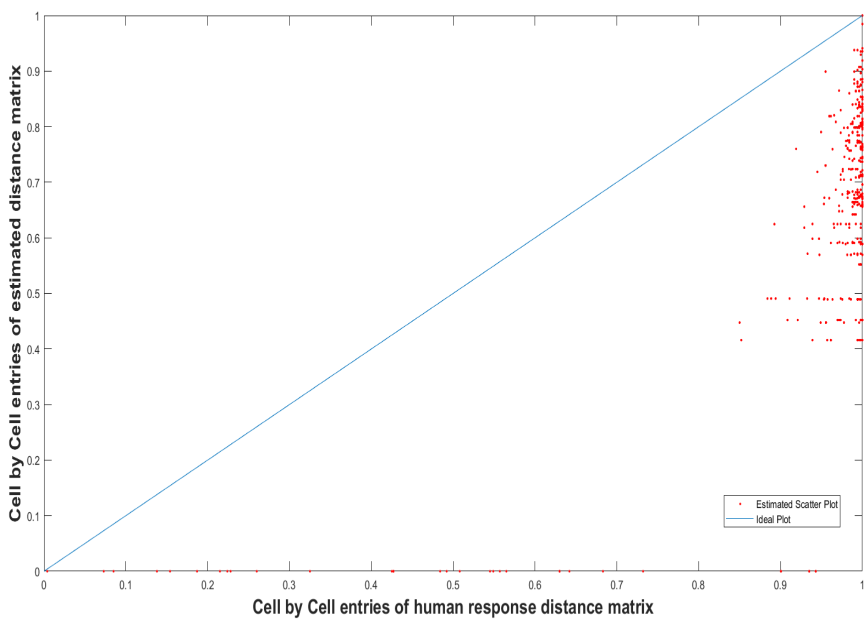
An accuracy check is done by finding the standard Euclidean norm between the similar and dissimilar pairs (information of similarity and dissimilarity obtained from T). Once the norms are found we check for how many of those the similar pair distance is lesser than the dissimilar pair. Mathematically, the accuracy is found out by the following expression:

$$\eta = \frac{\sum_{ijk} 1\{\|Lx_i - Lx_j\| < \|Lx_i - Lx_k\|\}}{n} \times 100$$

where n is the total number of triplets as defined above. A point to be noted here is that the signals x_i, x_j and x_k are not the same as the ones used for learning the distance metric (otherwise the accuracy would always be 100%). They are the signals other than the ones used for training out of the 10 samples collected for every material.

6.2 For the Braille Character Dataset

The accuracy of our algorithm is measured by how relatable our estimated distance matrix is to the human response distance matrix. There is no single formula which would give us this measure of closeness/accuracy. Rather, we have to analyze the relation using some graphical approaches. Ideally, the plot of the corresponding cell entries of the estimated distance matrix and the human response distance matrix must be a straight line of inclination of 45° with the allowance for some minor deviations. But the graph we obtain is:



The error of our estimate can be measured by how much our estimated data-points have deviated from the ideal plot. A way of doing this is to sum the perpendicular distances of our estimated points from the line. Mathematically it can be expressed as:

$$\epsilon = \sum_i \frac{|x_i - y_i|}{\sqrt{2}}$$

where (x_i, y_i) is the coordinate of the i^{th} estimated data-point.

7 Results

7.1 The Haptic DML algorithms

7.1.1 Algorithm 3

1. This algorithm gives us a very poor accuracy (almost close to 0%).
2. This value of such a low accuracy gives us another insight into the algorithm behavior. It shows us that instead of the desired operation *i.e.* the similar pairs must come closer and the dissimilar pairs must go apart, the exact opposite happens.
3. The probable reason to why this is happening :
   (a) Supposing there are 1000 triplets. Hence we have 1000 inequality constrains.
   (b) When we sum the constrains up-to the given iteration and then apply gradient ascent there is no guarantee that each of the individual constrains are satisfied.
4. It could so happen that one of the dot products (Ar, M) could be large enough (after applying gradient ascent), thus compensating for all the other inequalities.

7.1.2 Using the Standard CVX package

This approach using the standard library gives us a fairly good accuracy for the relaxed value of constrains, which is nothing but making $\varepsilon_{ijk} = 0$. The distance metric obtained for a random set of 500 triplets, using the standard CVX package, is given below:

$$\begin{pmatrix} 0.0713 & -0.0378 & 0.0303 & 0.0604 & 0.0575 & 0.0282 \\ -0.0378 & 0.1972 & -0.0065 & 0.0056 & 0.0171 & 0.0291 \\ 0.0303 & -0.0065 & 0.1935 & -0.024 & -0.0073 & 0.0234 \\ 0.0604 & 0.0056 & -0.024 & 0.1867 & -0.0192 & 0.0081 \\ 0.0575 & 0.0171 & -0.0073 & -0.0192 & 0.1957 & -0.0185 \\ 0.0282 & 0.0291 & 0.0234 & 0.0081 & -0.0185 & 0.1615 \end{pmatrix}$$

The accuracy results are given for different values of numerical error and for any randomly generated triplets (highest value of accuracy is tabulated after many trials of randomly generating triplets). The results are tabulated as follows

Accuracies versus the Numerical Error		
Numerical Error	Training Accuracy	Testing Accuracy
10^{-17}	50.5%	47.5%
10^{-16}	69.0%	63.2%
10^{-15}	99.0%	98.5%
10^{-14}	83.1%	80.8%
10^{-13}	100%	100%
10^{-12}	100%	100%

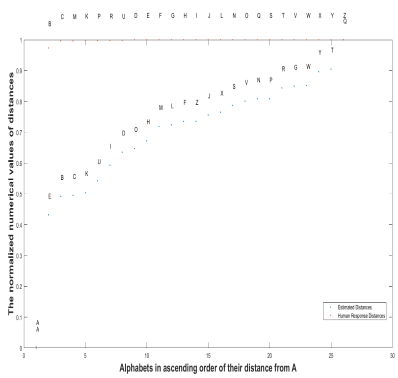
When the numerical error is increased above 10^{-12} the accuracy always comes out to be 100%. This gives us an estimate of the numerical limitation of the computational power.

7.2 The Least Squares Regression Algorithm - Braille Character Dataset

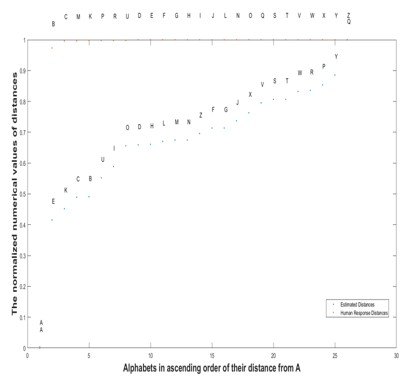
The estimated distance metric obtained using the least squares algorithm is the following:

$$\begin{pmatrix} 0.6623 & 0.2339 & -0.07276 & 0.2365 & 0.0196 & 0.0966 \\ 0.2339 & 0.4096 & 0.00317 & 0.0255 & 0.0196 & 0.06959 \\ -0.07276 & 0.00317 & 0.3505 & 0.00945 & 0.04466 & -0.0852 \\ 0.2365 & 0.0255 & 0.00945 & 0.417 & 0.01926 & 0.02378 \\ 0.0196 & 0.0196 & 0.04466 & 0.01926 & 0.2965 & -0.04125 \\ 0.0966 & 0.06959 & -0.0852 & 0.02378 & -0.04125 & 0.3461 \end{pmatrix}$$

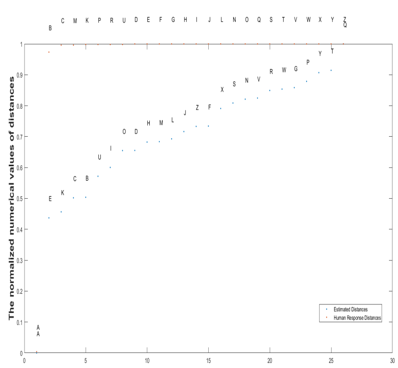
The closeness of the estimated distance matrix and the original human response distance matrix is given by a comparative study of the plots of the distance versus the letter.
The plots are shown below:



(a) Row Stacking



(b) Column Stacking



(c) U-Shaped Stacking

Figure 1: Plots of distance of alphabet A with all the others

Explanation of the plots shown above:

1. The X-axis depicts each letter of the English language spaced at equal intervals.
2. The order of letters is not alphabetical but rather the ascending order of the distances from the reference alphabet.
3. The reference alphabet in this case is the letter A.
4. The Y-axis depicts the numerical value of the distance of the reference letter with respect to the letter written above each of the data-point.
5. The Y-axis distance is normalized by dividing each of the entry in the estimated distance by the maximum value in the distance matrix to make all the entries between 0 and 1.
6. Normalization is done since our human response distance matrix's entries are also between 0 and 1, as it depicts the fraction of people.

7.3 A Different Confusion Matrix

An observation of the above plots (of the distance of each character with respect to others) suggests that the similarity perception of the braille alphabet could depend upon different aspects of the alphabets such as :

1. The number of elevated dots.
2. The orientation of the dots *i.e.* vertical or horizontal alignment.

Initially, we convert our 26×26 confusion matrix into a 5×5 confusion matrix by grouping the rows and columns suitably according to the groups. After that we sum all the entries of the corresponding group as well as one group with the other and place it in the corresponding columns of our new confusion matrix. After this we normalize each of the row of this new matrix so as to make all the entries sum up to 1. We symmetrisize this matrix. The new 5×5 confusion matrix(C_f) is given below:

# of dots raised	1	2	3	4	5
1	0.9960	0.0081	0.0007	0.0004	0.0000
2	0.0081	0.9047	0.0979	0.0211	0.0101
3	0.0007	0.0979	0.7107	0.2099	0.0844
4	0.0004	0.0211	0.2099	0.6584	0.4324
5	0.0000	0.0101	0.0844	0.4324	0.0000

After constructing C_f, we learn our distance metric M based on C_f and a set of new feature vectors(X_g) which consists of the pairwise distances of one alphabet group with itself and also one alphabet group with the other. The matrix shown below is for row stacking feature vectors:

# of dots raised	1	2	3	4	5
1	1.0000	0.0081	0.0007	0.0004	0.0000
2	0.0081	0.9083	0.0983	0.0212	0.0102
3	0.0007	0.0983	0.7136	0.2108	0.0847
4	0.0004	0.0212	0.2108	0.6611	0.4341
5	0.0000	0.0102	0.0847	0.4341	0.0000

8 Future Work

For testing another approach to the problem we are going try the boost metric approach as given in the work of Shen *et.al.*[6]. Another possible approach is to try to learn te matrix L instead of M since then the condition of positive semi definiteness need not be checked as $L^T L$ will automatically be PSD.
After finding some robust distance metric learning algorithms, the concept of DML could be applied to different types of data.
I had worked on a project on brain tumor detection and classification on MRI scans. We had applied a CNN for classifying the tumor into malignant and benign. A DML approach could be applied to the same problem where the 2 different types of tumors are first clustered and then a machine learning algorithm is applied. This would result in a much better accuracy of the machine learning algorithm.

9 Acknowledgment

I would like to thank the Indian Academy of Sciences for providing me the opportunity to intern at the Indian Institute of Technology, Bombay, the most prestigious technical institute in India. I would also like to thank Ms. Priyadarshini for all her support and guidance. I would like to thank Prof. Subhasis Chaudhuri for giving me the opportunity to work in the Vision and Image Processing Lab. Lastly, I would like to thank all the doctoral and masters students in the lab for their continuous guidance .

References

[1] E.P. Xing, A.Y Ng, M.I. Jordan, and S. Russel, *Distance Metric Learning with Application to Clustering and Side Information*, NIPS, 2003, pp. 521 - 528

[2] J.M. Loomis, *Tactile recognition of raised characters: A parametric study*, Bulletin of the Psychonomic Society, 1985,Vol 31, No.2, pp. 18 - 20

[3] M. Strese, C. Schuwerk, A. Iepure, and E. Steinbach, *Multimodal Feature-based Surface Material Classification*, IEEE Transactions on Haptics, Vol.10, No.2, pp. 226 - 229, 2017

[4] N. Landin, J. M. Romano, W. McMahan, and K. J. Kuchenbecker, *Dimensional Reduction of High-Frequency Accelerations for Haptic Rendering,*, International Conference on Human Haptic Sensing and Touch Enabled Computer Applications,Springer, Vol.x ,No.x , pp. 79 - 86,2010

[5] S.P. Boys and M.C. Grant, *The CVX Users Guide*, Stanford University, Palo Alto, CA.

[6] C. Shen, J. Kim, L. Wang, and A. Hengel, *Positive Semidefinite Metric Learning with Boosting*, NIPS, Vol. , No. , pp. 1651 - 1659, 2009

[7] S. Bensmaia and M. Hollins, *Pacinian Representations of Fine Surface Texture,* Perception & Psychophysics, Vol. 67, No. 5, pp. 842 - 854, 2005.