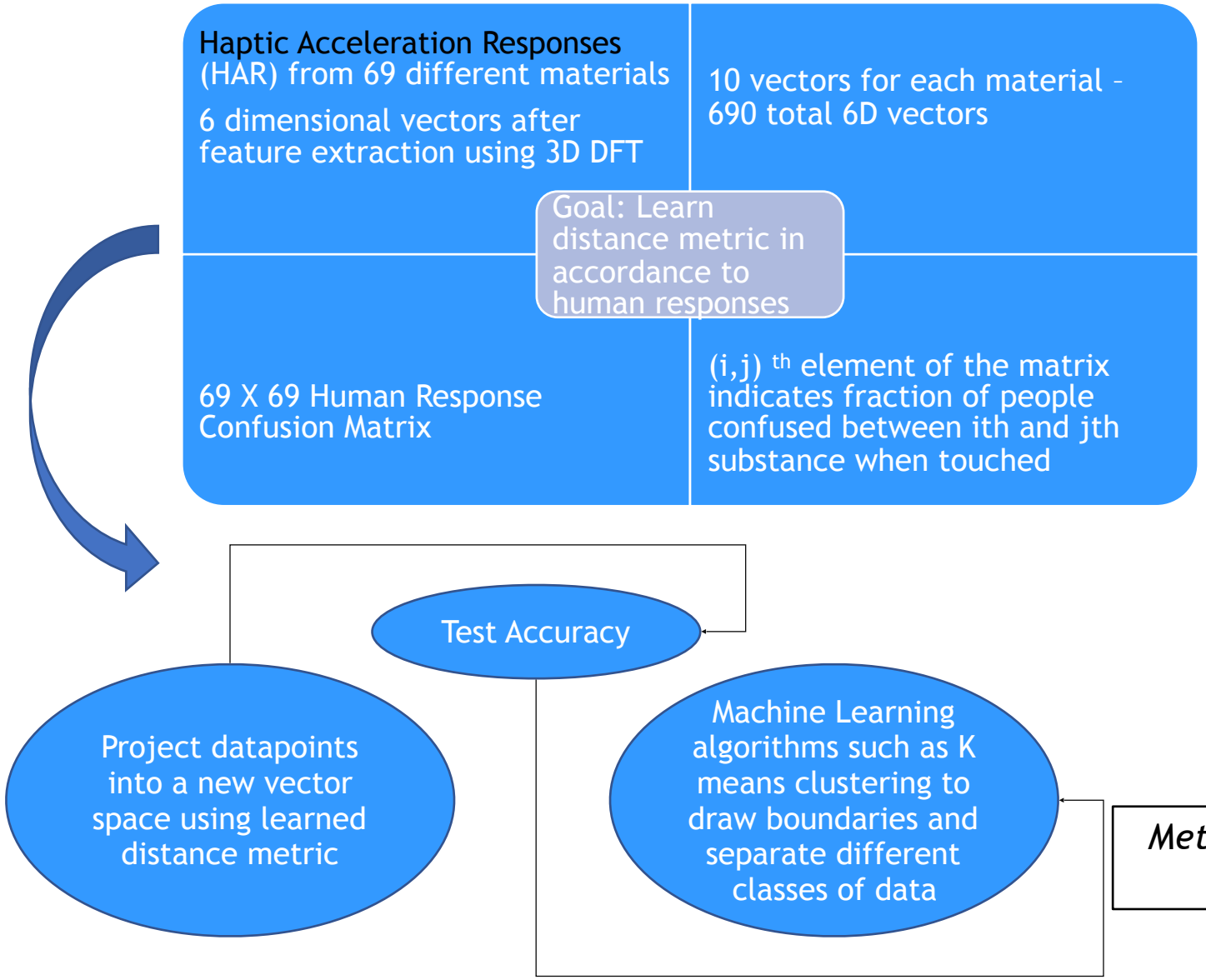
Distance Metric Learning for Haptic Data Classification

Megh Bhalerao

3rd Year B.Tech., Electrical & Electronics Engineering

National Institute of Technology Karnataka, Surathkal, Mangalore, India

HAR Solution using Constrained Convex Optimization



- All Possible Triplets (i, j, k) generated using the confusion matrix such that C(i, j) > C(j, k)
- Constrained optimization problem formulated with such constrains - distance calculated between feature vectors using distance metric must be in compliance with the triplets obtained

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

where

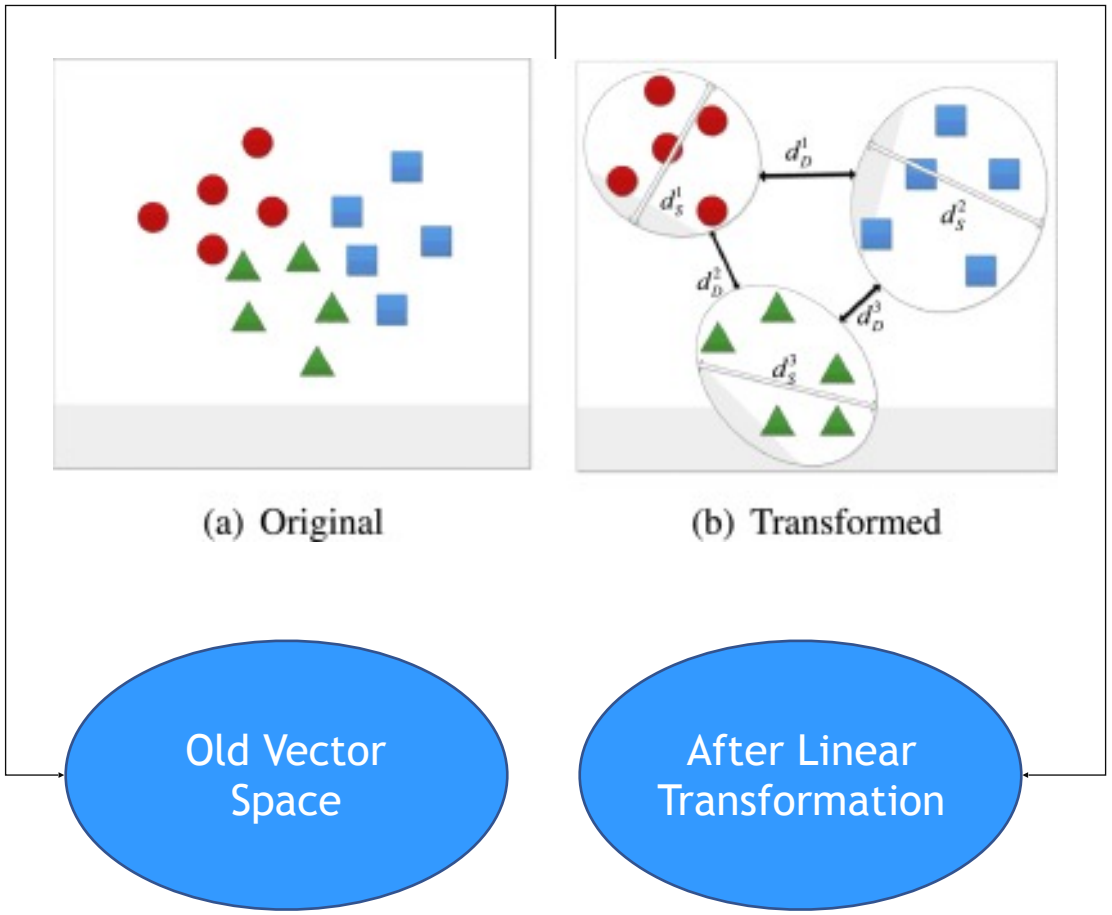
$$Ar = (x_i - x_j)(x_i - x_j)^T - (x_i - x_k)(x_i - x_k)^T$$

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

- Solution obtained using gradient descent algorithm

Metric Learning when applied prior to ML / DL drastically increases the accuracy

HAR Solution Accuracy Measurement

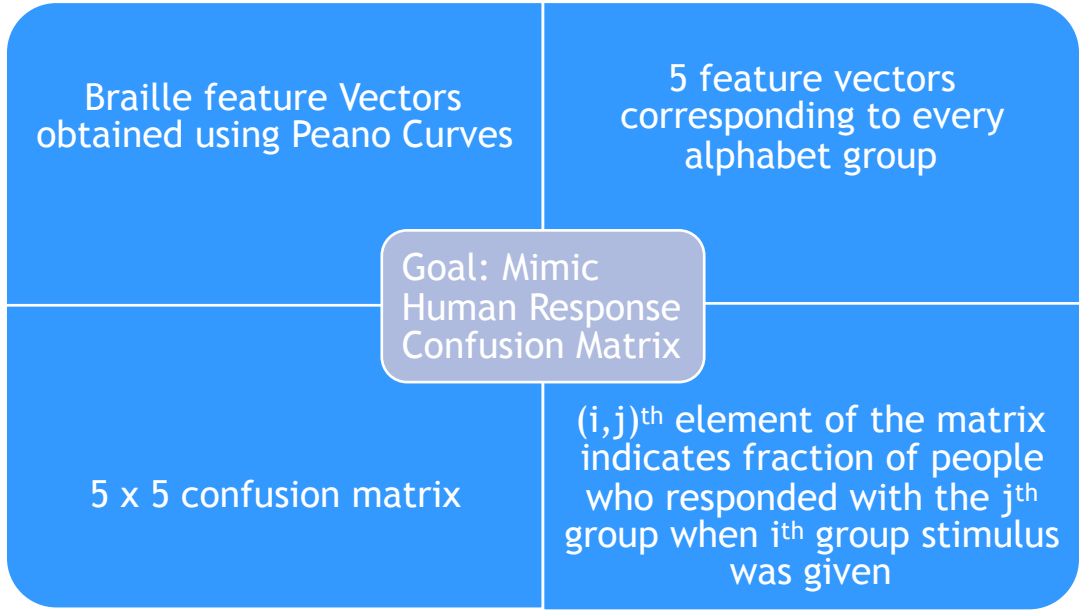


- Distance metric accuracy is measured by number of triplet conditions satisfied when the set of feature vectors is projected into a new vector space using our learned distance metric
- Accuracies for various numerical error shown :

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%

As the numerical error constrain gets more relaxed the accuracy is seen to increase

Braille Character Dataset Application



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

Project datapoints in a new vector space → Re-estimate confusion matrix by calculating pairwise distances between alphabets' feature vectors → Compare original and estimated confusion matrix

Given Distance Matrix – Entries according to Human response

# of dots raised	1	2	3	4	5
1	0.0040	0.9919	0.9993	0.9996	1.0000
2	0.9919	0.0953	0.9021	0.9789	0.9899
3	0.9993	0.9021	0.2893	0.7901	0.9156
4	0.9996	0.9789	0.7901	0.3416	0.5676
5	1.0000	0.9899	0.9156	0.5676	1.0000

Estimated Distance Matrix – Entries as calculated by braille feature vectors and learnt distance Metric

# of dots raised	1	2	3	4	5
1	0	0.1541	0.9632	1.0169	1.1209
2	0.1541	0.4821	0.7339	0.8043	0.9567
3	0.9632	0.7339	0.3967	0.7911	0.9834
4	1.0169	0.8043	0.7911	0.3295	0.8694
5	1.1209	0.9567	0.9834	0.8694	0.5357

Classical optimization doesn't work up to the mark (as seen from entries of given and estimated distance matrices). Neural Network approaches could give better results.

Thank You!