Advanced Techniques in Machine Learning
Ex 1
Eyal Orbach - ID 015369317
Daniel Juravski - ID 206082323

Below is a table showing the 'correct prediction' percentage on the validation data (20% of the train data that was not trained on)

distance function\comparison type	One vs All	All Pairs	Our own (tournament)
Hamming	92.97%	96.68%	93.81%
Loss	95.67%	96.84%	94.32%

Our own matrix:
We Implemented a "tournament" like check.
we created 3 weight vectors:

1) distinguishes between classes 0 or 1 vs classes 2 or 3
2) distinguishes between class 0 vs class 1
3) distinguishes between class 2 vs class 3

vector 1 trained on all examples while vectors 2 and 3 only on the classes it can distinguish between.

then our m matrix was so:

$$\begin{array}{r} class0 = \\ class1 = \\ class2 = \\ class3 = \end{array} \begin{pmatrix} 1 & 1 & 0 \\ 1 & -1 & 0 \\ -1 & 0 & 1 \\ -1 & 0 & -1 \end{pmatrix}$$

where each row is compared to the results vector (fx1, fx2, fx3)

This produced ok results, better than Hamming with One vs All, but not as good as the rest.

Role of distance in rows

Hamming distance checks which of the vectors received the same sign as the equivalent index in the relevant class' row and which didn't. The less different signs the closer that vector is.

Loss distance simply adds the relevant loss function (svm hinge loss) for each row, where the significance of fx is determined by the equivalent index in the m matrix. 1 when we wish this vector will classify it positively, -1 if we this vector classifies it negatively and 0 when we do not care about the result

<u>One vs All:</u>
Here we have noticed the most significant difference between Hamming and loss distance. This can be explained because the loss distance outperforms the Hamming distance in these 2 use cases:
-) The relevant vector did not classify correctly the item, but the other vectors have strongly classified it as not belonging to their class. In this case Hamming will have a tie of all values equal to 2.5, resulting in predicting class 0 due to our tie breaking rule.
Where as the loss distance will recognize the significance of the other vectors negative values.
-) More than one vector classified with a positive value but not with a large degree of confidence (a value smaller than 1). the Hamming distance again will tie between those classes where the loss distance can determine by the level of confidence (values above 1 will still tie though)

<u>All pairs:</u>
Here the advantages of the Loss distance are less significant because Hamming has produced nice results also. this is due to the larger granularity of comparison.
There are 6 vectors 0vs1, 0vs2, 0vs3, 1vs2, 1vs3, 2vs3.
the rows in the m matrix determine which vectors it requires a positive value and which a negative value in order to classify it as this class. only the 3 vectors that correlate to the class are checked the rest contribute a fixed value due to having 0 in the m matrix.

conclusions:

- The All Pairs with Loss distance function has produced the best results
- The biggest difference is in 'One vs All' where loss distance gives much better results than hamming distance.
- Loss distance dominates Hamming on all expirements