CS 559: Final
Duration: 2:30

Answer all problems. Show all calculations and provide sufficient explanation. If in doubt, explain more.

Short and to-the-point answers are preferred.

NAME: ARCHANA KALBURGI

CWID: 10469491

Problem 1. (30 points) True of False. If you choose false, briefly explain.

True
- SVMs can be used in an ensemble classifier. (True or False) TRUE

- it can be used as one of the classifiers

False
- In bagging, we choose random samples from the input without replacement. (True or False) False

- we replace the samples

True
- A deep neural network with a non-linear activation function in each layer is meaningless. (True or False) TRUE

- purpose of activation function is to introduce non linearity into the ntw

False
- Linear discriminant functions always lead to non-convex decision regions. (True or False)

Decision regions are convex in LDF

False
- To achieve better performance, we should look at the testing set when training a classifier. (True or False)

Test set should always be set apart

True
- When applying K-Means for clustering, the distance measure should always be differentiable. (True or False)

Because we set it to zero.

True
- Compared to SVM, Adaboost is more sensitive to and thus easier to fail on un-normalized data. (True or False)

In SVMs data features have to be normalised.

True
- A kernel matrix have to be designed to be invertible. (True or False)

invertible ≈ non-singular.

False
- Using high-degree polynomial features help to reduce overfitting in a SVM classifier. (True or False)

∵ with increased dimension :- flexible decision boundary

False
- EM algorithm always finds the global optimal. (True or False)

Not gauranteed

False

- Deep neural networks usually require fewer training data as compared to Adaboost. (True or False)

Adaboost work fine with less training data
DNN need more data

True

- When training a random forest, the order of training the decision trees does not play a role. (True or False)

Order does not matter.

True

- When applying SVM, we have to explicitly specify a kernel function that maps a sample from a lower dimension to a higher dimension. (True or False)

In SVM we need to choose the Kernel

True

- Deep neural networks rely on setting the weights of the networks mannually.

we stat with manually setting weight.

False

- In Adaboost, the weak learner that yields a higher error rate results in a higher weight in the combined classifier.

weak learner with low error Rate has
larger weight

Problem 2. (20 points) In AdaBoost, the weight of sample x_i in round t is updated according to:

$$D_{t+1}(i) = D_t(i) \cdot exp(-\alpha_t y_i h_t(x_i))$$

where $y_i \in \{-1, 1\}$ is the label of sample x_i, $h_t(x_i)$ is a weak learner that returns $+1$ or -1 depending on which class it predicts for x_i and α_t is the (positive) weight of the weak learner.

(a) Explain under which conditions the weight of a sample is increased or decreased. (5 points)

(b) How does this mechanism of updating the weights affect the selection of subsequent weak learners? (5 points)

(c) Initially, all samples have equal weights. In the first iteration, we use a weak learner $h_1(x)$ on the data, update the weights and normalize them to sum to 1. Which of the following weight distributions, both sum up to 1, is NOT valid and why? (10 points)

1. $D_2 = \{0.1667 \;\; 0.25 \;\; 0.25 \;\; 0.1667 \;\; 0.1667\}$
2. $D_2 = \{0.1538 \;\; 0.2308 \;\; 0.2308 \;\; 0.1538 \;\; 0.2308\}$

a) weight of the weak learner is increased when x_i is misclassified conversely correctly classified would recieve decreased weight.

b) The assigned wt to the weak learners increases or Decreases based on correctly or incorrectly classified Samples. So the final classifier will be the weighed combination of weak learners.

c) (2) option is the valid one because it has. to update the weight we have to choose the low weighed error. For Ex: let error rate $= x$ initial weight $= 0.2 (1/5)$.

$0.2308 = 0.2x$ VS $0.25 = 0.2x$

(2) $x \cong 1.154$ (1) $x = 1.25$

$(2) < (1)$

Problem 3. (15 points) Given a set of 2D data $X = \{[-4\ 1], [-4\ -1], [1\ 1], [1\ -1], [3\ 1], [3\ -1]\}$, we apply the k-means algorithm to cluster them into two clusters. Starting from two different initializations, we obtain two solutions represented by their centroids:

Sol. 1: $C_1 = [-4\ 0]$ and $C_2 = [2\ 0]$

Sol. 2: $C_1 = [0\ 1]$ and $C_2 = [0\ -1]$

How would you select which of the two solutions to use? Which clustering is preferable and why? It is safe to assume that the algorithm has converged in both cases.

Sol

	1) Distance from centroid 1 [-4,0]	Distance from C2[2,0]
1) [-4,1] :	1	6.082
2) [-4,1] :	1	6.082
3) [1,1] :	5.09902	1.414
4) [1,-1] :	5.09902	1.414
5) [3,1] :	7.071068	1.414
6) [3,-1] :	7.071068	1.414
	(1)	(1.414)

Sol 2) Distance from C1 [0,1]	Distan from C2 [0,-1]
4	4.472
4.47	4
1	2.236
2.236	1
3	3.605
3.605	3
(1)	(1)

- For good cluster : internal distances should be small & external should be large.
- I would pick solution 1.

After assigning all the closest points to centroid sample1 & sample2 belong to C1 group. & the rest belong to C2 & there is a clear separation seen eg, sample 1 & 2 has min dis for C1 group but also has max distance from C2 vice versa for other samples

Problem 4. (15 points) Consider the k-means clustering algorithm which comprises the following steps:

1. Initialize by selecting k cluster centers arbitrarily and assign each example to the closest center.
2. Compute sample means for each cluster.
3. Reassign all samples to the closest mean.
4. If clusters changed at step 3, go to step 2.

The objective function of the k-means algorithm is the sum of squared errors:

$$J_{SSE} = \sum_{i=1}^{k} \sum_{x_j \in D_i} ||x_j - \mu_i||^2$$

where x are the samples, D_i is the i^{th} group and μ_i is its center.

Explain how the objective function is minimized by step 2 of the k-means algorithm and, then, how it is further minimized by step 3.

step 2 of K-means is assinging data pts to cluster.
lets say the assignment of sample j to a cluster c is denoted by C_j then the sum of squared distance betn samples & cluster means. can be written as,

$$\text{argmens} \sum_{i\in 1}^{K} \sum_{x_j\in D_i} ||x_j - u_i||^2 = \sum_{j=1}^{n} ||x_j - u_{c(j)}||^2$$

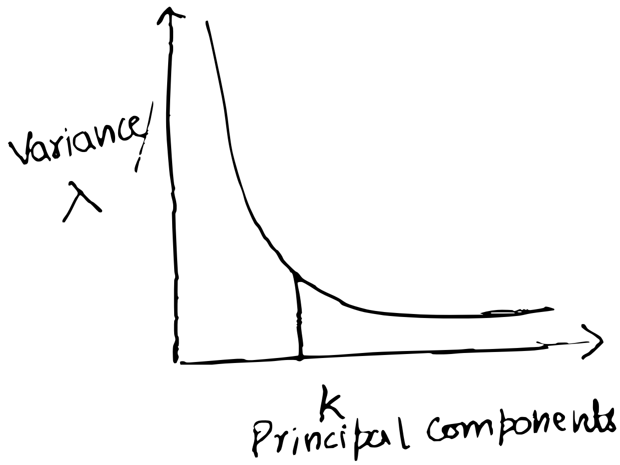
step2 Minimises the objective fun by fixing u. It is exactly assigning each sample to the closest cluster

objective fun is minimised by step 3 by fixing C & finding the best mean for each cluster.
Best mean is obtained by setting u_i to the mean of samples assigned to this cluster.

Problem 5. (10 points) You have been given 1000 samples from each of two classes, ω_1 and ω_2, in 100-D and you wish to find the number of principal components that should be kept in order to achieve the maximum possible accuracy using a linear SVM after PCA. Consider all parameters of the linear SVM fixed.

If k denotes the number of principal components kept (and 100-k the number of principal components discarded) that result in good classification performance, describe the procedure you would follow in a way that it can be implemented by someone who is not familiar with PCA or the classifier, but can implement all necessary data structures and store intermediate results if necessary.

→ • Find Principal components for the data.



- find PC with maximum projection variance.

$$V = v^T A v$$

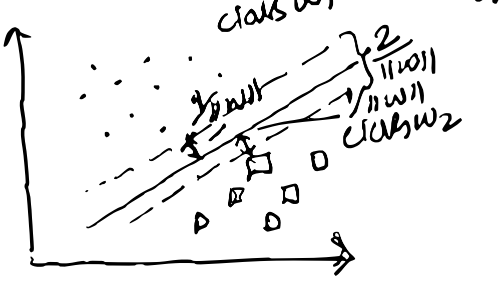
$$A = \sum_x (x-\bar{x})(x-\bar{x})^T$$

V_i : eigenvector of A with largest eigenvalue.

PC : eigenvector with highest eigenvalue.

Above graph helps us pick the optimal number of PCs where K is number of principal component & λ or the variance captured by the PCs. PCs are stored as new vectors.

Project the data back to the original subspace and implement SVM on the projected data. In order to apply SVM. Initialise the 2 closest points of different class and maximise the distance between them to get the svm. $g(x) = w^t x + w_0$



max distance betn 2 points is given by

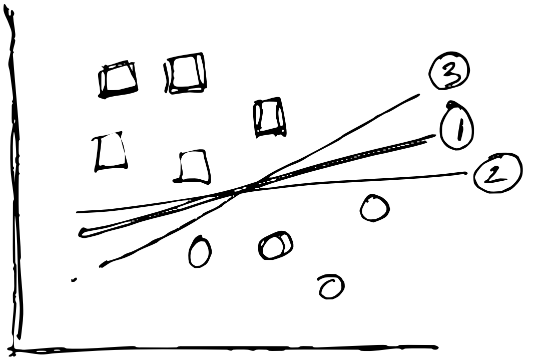
$$\frac{w^t x + w_0}{\|w\|} = \frac{1}{\|w\|} \qquad \text{margin} = \frac{2}{\|w\|}$$

Problem 6. (10 points) Consider the problem of training a classifier on a training set comprising two **linearly separable** classes. Explain why maximizing the margin using a Support Vector Machine is a better option than

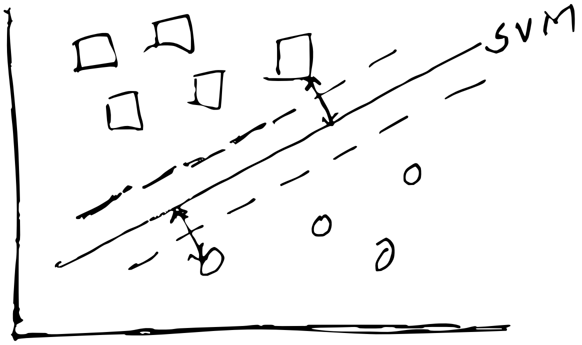
(a) minimizing the number of misclassified samples using a Perceptron (5 points)

(b) training a classifier using Maximum Likelihood Estimation and modeling the classes as multi-variate Gaussian distributions (5 points)

a)



Perceptron



SVM

Perceptron gives one of the many possible solution SVM gives one optimal solution to classify the two linearly separable classes. If the test sample is close to the classifier, the unknown sample might get wrongly classified. SVM avoid this prob

b) MLE does not take the advantange of linear separablity. It is based on Bayesian theory in estimating parameters. of problistic model. Where as SVM is an optimization based on nonparametic method.

MLE does not give good generalization like SVM