CS 391L Homework 3 - Theory
Machine Learning

1) A be an $m \times d$ matrix

let $X = AA^T$

Assume, X has d distinct, non-zero eigen values
Assume, $m \gg d$.

Find the eigendecomposition of X, we'll need to find the eigendecomposition of an $m \times m$ matrix. But since m is much larger than d, this is slow.

Algo :- that only requires computing the eigen decomposition of a $d \times d$ matrix

$$\Longrightarrow \quad X = AA^T$$
$$Y = A^T A \in R^{d \times d} \quad (\text{a } d \times d \text{ matrix } A^T A)$$

Use eigendecomposition to Y

$$Y = \sum_{i}^{d} \lambda_i V . V^T$$

λ_i eigenvalues of $Y = A^T A \in R^{d \times d}$
v_i eigenvalues of Y

From S.V.D we know $A = U . S V^T$ so
$$Y = (VSU)(VSU)^T$$
$$Y = VS^2V^T$$

$$x = (usv)(usv)^T = us^2u^T$$

We have $\lambda_i = s^2$ as the eigenvalue of Y.

λ_i also can be considered as the eigenvalue of X.

use $\sum_i^m \lambda_i u \cdot u^T$ to calaulate the eigenvector of X. The first d elements same as $\lambda_1, \ldots \lambda_d$

$m \gg d$, the left $\lambda_{d+1} \ldots \lambda_m$ are equal to zero

Alternatively, with a $Y = A^T A$, with $\bar{w}$ as eigenvector of Y with eigenvalue λ_w, s.t. $Y\bar{w} = \lambda_w \bar{w}$ plugging in A, we get $(A^T A)\bar{w} = \lambda_w \bar{w}$, multiplying both sides by A $AA^T A\bar{w} = A\lambda_w \bar{w}$, since $X = AA^T$ $XA\bar{w} = \lambda_w A\bar{w}$, which matches our definition of eigenvector eigenvalue and is our eigendecomposition of X where $(A\bar{w})$ is the eigenvector and λ_w is the eigenvalue. Thus we can find the eigendecomposition of X through the following

1. Find the eigendecomposition of Y where $Y = A^T A$ using our "black box" procedure
2. The eigenvalues of $Y(\lambda_w)$ is the same as the eigenvalues of X. The eigenvectors of $X(w)$ can be found multiplying the eigenvector of $Y(\bar{w})$ by A i.e. $w = A\bar{w}$

2.

a) True
b) True
c) False

d) multi collinearity (dependent input vars)

Use PCA to reduce the dimensionality of the input data in the effort to collapse correlated variables into a single variable. This should then leave us with a set of features that are largely uncorrelated with one another. After the principal components are found, you can transform your original matrix by projecting the matrix onto the principal components. This is essentially principal component regression. Using the principal components, we capture the entirety of the dataset with zero error.