Machine Learning HW3 - Theory

1. **Question:** Let A be an $m \times d$ matrix, and let $X = AA^T$. Assume that X has d distinct, non-zero eigenvalues. Assume that $m >> d$. In order to find the eigendecomposition of X, we will need to find the eigendecomposition of an $m \times m$ matrix. Since m is much larger than d, this is slow. Give an algorithm for finding the eigenvectors and eigenvalues of X that only requires computing the eigendecomposition of a $d \times d$ matrix.

Answer: So we want to find eigenvectors Q and eigenvalues D such that $AA^T = X = QDQ^T$, knowing that D holds d values λ_i.

Consider the SVD for $A = USV^T$. $AA^T = USV^T(USV^T)^T = USS^TU^T = QDQ^T$, and S has d diagonal values σ_i such that $\sigma_i^2 = \lambda_i$ (these results are also discuss in class and in the Kalman reading).

Let's now look at A^TA, a $d \times d$ matrix. $A^TA = (USV^T)^TUSV^T = VS^TSV^T = VD'V^T$. Thus $D' = S^TS$ is a $d \times d$ matrix with d diagonal values $\sigma_i^2 = \lambda_i$. So if we have D', we in essence have the eigenvalues we need (you can think of D as being D in the top-left quadrant, and 0 everywhere else). And $A = USV^T \iff US = AV$. Thus $u_i\sqrt{\lambda_i} = Av_i \iff u_i = \frac{1}{\sqrt{\lambda_i}}Av_i$. So that's how we can find the eigenvectors.

Note that only the first d eigenvectors of U are relevant the rest of U map to the null-space.

Putting it all together for our algorithm A:

```
def eigen(A):
  A_T = transpose(A)
  V, D_prime = eigendecompose(A_T * A)

  for i in 1..d
    u_i = 1/sqrt(D_prime[i])*A*V[i]
  end

  U = [u_1, .. u_d]

  return U, D_prime
```

2. **Question:** In this problem we explore some relationships between SVD, PCA and linear regression.

(a) True or false: linear regression is primarily a technique of supervised learning, i.e. where we are trying to fit a function to labeled data.

Answer This is true. With linear regression we want to find a vector x such that we minimize $|Ax - b|$. In other words, the objective is to find a linearly dependent relationship between a dataset with d features and the labels b. The labels have to be known ahead of time.

You could apply linear regression to also to discover relationships between existing data, but that can be done more generally via PCA and SVD.

(b) True or false: PCA is primarily a technique of unsupervised learning, i.e. where we are trying to find structure in unlabeled data.

Answer This is also true. With PCA, you can discover relationships in your data, and correlations between particular features. The objective of PCA is to find an orthonormal basis that maximizes variances in a linear model, thus extracting particular values and vectors that "most represent" the data or hidden linear variables therein.

(c) True or false: SVD is primarily an operation on a dataset whereas PCA is primarily an operation on a matrix.

Answer I would say this is false and it's the other way around (although you can think of them doing both). We do think of PCA in the context of a dataset, trying to project it on the most significant dimensions, and it is implemented via an eigendecomposition (which is a matrix operation on the covariance matrix of the input). SVD is a linear algebra result and technique that applies to any matrix; it can replace eigendecomposition as a means to do PCA, but it also has other applications (it can be applied to "datasets" to compress them).

(d) A common problem in linear regression is multicollinearity, where the input variables are themselves linearly dependent. For example, imagine a healthcare data set where height is measured both in inches and centimetres. This is a problem because there may now be multiple w satisfying $y = w \cdot x$. Explain how you could use a preprocessing step to solve this problem.

Answer You can use PCA via SVD to preprocess the data in order to remove linearly dependent columns (redundant data). Assume the data is of size $m \times d$, $m > d$. If there are linearly dependent columns in our matrix A then $A = USV^T$ will have rank less than d; S will have 0 entries on its diagonal, and U will tell us which columns in A map to the 0 values (the null-space) and are therefore unnecessary. We can then ignore the columns from our dataset.

We could even use $AV = US$ instead of our dataset to get a "compressed" version of the dataset; it would remove both redundant columns and rows, but, depending on what we're doing next in the process, this may not be desirable since it changes the data.