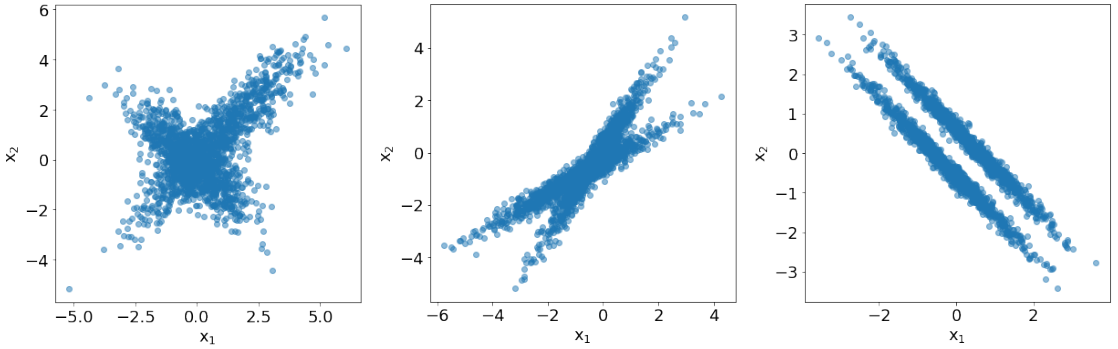
This homework is due on Thursday, November 19, 2020, at 10:59PM.

Self-grades are due on Thursday, November 26, 2020, at 10:59PM.

1 Implementation: SVD and PCA

In this problem we will implement Principal Component Analysis (PCA) using Singular Value Decomposition (SVD) in python.



(a) Dataset 1 (b) Dataset 2 (c) Dataset 3

Figure 1: Example datasets that we will work with.

Figure 1 shows the three datasets that we will be working with. Each dataset is comprised of entries $\vec{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$. We will develop a procedure to calculate the principal components $\vec{v}_i$ and corresponding weights w_i for this dataset by calculating an SVD of the matrix A. Use the supplied iPython notebook to complete this problem.

a) Consider the following datasets. First, intuitively sketch the Principal Components (PCs) that best explain this dataset. You don't need to calculate the exact lengths of the principal components but their relative lengths should approximately be correct.

b) We are given a set of data points $\vec{x} \in \mathbb{R}^2$. Does our data have 0 mean? If not, subtract the mean from this data and construct a new *de-meaned* matrix $\hat{A}$.

c) Explain the relationship between

   i) Diagonalization of the sample covariance matrix $C = \frac{1}{N}\hat{A}^T\hat{A}$,

   ii) Singular Value decomposition of $\hat{A}$, and

   iii) Principal Components of $\hat{A}$.

   In class, we have learnt that we can calculate the SVD (and thereafter PCA) of the matrix A using the sample covariance matrix C. However, this approach is not used in practice for numerical reasons. Some of the advanced classes on numerical methods delve deeper into this topic. We will be using the inbuilt SVD-solver in python to get the SVD of our de-meaned matrix $\hat{A}$.

d) What are the dimensions of the matrix $\hat{A}$? If we were to compute the full-svd, what will be the dimensions of matrices U, Σ, and V if the SVD is written as

$$\hat{A} = U\Sigma V^T \tag{1}$$

e) Compute the principal components (unit vectors) and corresponding weights for the datasets. Sketch them overlaid on the data (The weights have been scaled in the notebook to make visualization neater). Does this match what you had expected at the beginning of the problem? Comment on the differences in the principal components of the three datasets.

2 The Moore-Penrose Pseudoinverse for "Fat" Matrices

Suppose that we have a set of linear equations described as $A\vec{x} = \vec{y}$. If A is invertible, we know that the solution is $\vec{x} = A^{-1}\vec{y}$. However, what if A is not a square matrix? In EE16A, you saw how this problem could be approached for tall matrices A where it really wasn't possible to find a solution that exactly matches all the measurements. The linear least-squares solution gives us a reasonable answer that asks for the "best" match in terms of reducing the norm of the error vector.

This problem deals with the other case — when the matrix A is short and fat. In this case, there are generally going to be lots of possible solutions — so which should we choose and why? We will walk you through the **Moore-Penrose pseudoinverse** that generalizes the idea of the matrix inverse and is derived from the singular value decomposition.

a) Suppose that you have the following matrix.

$$A = \begin{bmatrix} 1 & -1 & 1 \\ 1 & 1 & -1 \end{bmatrix}$$

Calculate the full SVD decomposition of A. That is to say, calculate U, Σ, V, such that

$$A = U\Sigma V^T, \text{ where } U \text{ and } V \text{ are unitary matrices.}$$

Leave all work in exact form, not decimal.

Note: Do NOT use a computer to calculate the SVD.

b) Let us think about what the SVD does. Let us look at matrix A acting on some vector $\vec{x}$ to give the result $\vec{y}$. We have

$$A\vec{x} = U\Sigma V^T\vec{x} = \vec{y}.$$

Observe that U and V^T are unitary matrices, so they cannot change the norm of the input vector while Σ scales the input vector. We will try to "reverse" these operations one at a time and then put them together.

If U performs some transformation on the vector $\left(\Sigma V^T\right)\vec{x}$, what is the inverse of U that cancels its effect.

c) Recall that Σ has the same dimensions as A (m by n with $m < n$). Now find some *diagonal* $\widetilde{\Sigma}$ that "inverts" Σ. That is $\widetilde{\Sigma}\Sigma = \widetilde{I}_{n\times n}$ where $\widetilde{I}_{n\times n}$ given below is a block diagonal identity matrix with the top left block being an m by m identity matrix. You may assume that A has rank m. By diagonal for some non square matrix, we mean that the top (for thin matrix) or left (for wide matrix) square submatrix to be diagonal and zeros for the remaining submatrix.

$$\widetilde{I}_{n\times n} = \begin{bmatrix} I_{m\times m} & 0_{m\times(n-m)} \\ 0_{(n-m)\times m} & 0_{(n-m)\times(n-m)} \end{bmatrix}$$

d) What is the inverse of V^T that cancels its effect.

e) Try to use the previous parts to derive an "inverse" (which we will use $A^\dagger$ to denote). That is to say,

$$\vec{x} = A^\dagger \vec{y}.$$

The reason why the word inverse is in quotes (or why this is called a pseudo-inverse) is because we're ignoring the "divisions" by zero.

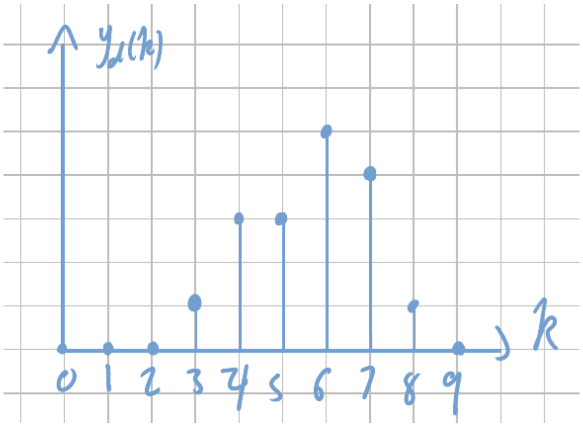
f) Use $A^\dagger$ to solve for $\vec{x}$ in the following system of equations.

$$\begin{bmatrix} 1 & -1 & 1 \\ 1 & 1 & -1 \end{bmatrix} \vec{x} = \begin{bmatrix} 2 \\ 4 \end{bmatrix}$$

3 Piecewise Linear interpolation

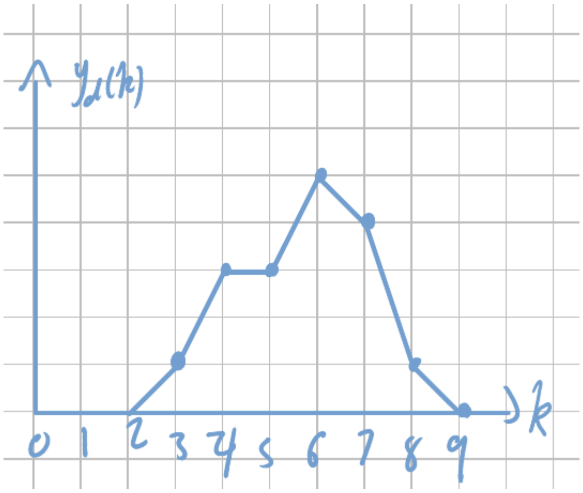
Suppose we have a discrete-time signal $y_d(k)$ that we would like to interpolate.

We will assume that the discrete-time signal is of *finite duration*– that is, the signal "begins" at some time k_1 and "ends" at some time k_2, and we can assume $y_d(k) = 0$ for $k < k_1$ and $k > k_2$. For example, if our discrete-time signal looked like this:



then we would have $k_1 = 3, k_2 = 8$.

One of the simplest ways to interpolate this signal would be to simply connect the points of the discrete-time signal with straight lines. If we were to interpolate the signal above this way, we would get this:



As you can see, the interpolated signal $y(t)$ is a straight line over intervals of the form $[k, k+1]$ for all integers k, although the entire function $y(t)$ is not itself a straight line. For this reason, we call this $y(t)$ the *piecewise linear* (PWL) interpolation of the discrete-time signal $y_d(k)$.

Although we've just described the PWL interpolation in an intuitive and somewhat *ad hoc* way, it turns out that the PWL interpolation can be expressed as a basis function interpolation. **in this problem, you will show how this is true**.

a) Consider the function $\phi(t)$ defined as,

$$\phi(t) = \begin{cases} 1 - |t|, & t \in [-1, 1] \\ 0, & \text{otherwise} \end{cases}. \tag{2}$$

Sketch $\phi(t-k)$ for some arbitrary integer k. You may choose a specific integer for k in your sketch ($k = 3$ perhaps), or you may keep the sketch entirely in terms of k. The graph in the solution will be drawn in terms of k, so we encourage you to try it that way.

b) We'll be using the function $\phi(t)$ as the basis function for the PWL interpolation.

We will begin our analysis right at the beginning of the signal. Write the basis function and coefficient that captures the line of $y(t)$ from $t = k_1 - 1$ to $t = k_1$. That is to say, find real number α and integer p such that,

$$y(t) = \alpha\phi(t-p) \text{ for } t \in [k_1 - 1, k_1]$$

c) Now, consider any integer $k^\star$such that $k_1 < k^\star < k_2$. Over the interval $[k^\star, k^\star + 1]$, the interpolated signal $y(t)$ is a straight line. What is the equation of this line? In other words, **find real numbers m and b such that**

$$y(t) = mt + b, \text{ over the interval } [k^\star, k^\star + 1]. \tag{3}$$

d) Consider the function

$$g(t) = y_d(k^\star)\phi(t - k^\star) + y_d(k^\star + 1)\phi(t - (k^\star + 1))$$

This function is also a straight line over the interval $[k^\star, k^\star + 1]$. What is the equation of the line over this region? Write it again in the form

$$y(t) = mt + b, \text{ over the interval } [k^\star, k^\star + 1]. \tag{4}$$

This should match your previous answer.

e) Given what you've shown in the previous parts, we can now express the PWL interpolation of $y_d(k)$ as a sum of shifted ϕ functions. Find the coefficients α_k such that

$$y(t) = \sum_{k=-\infty}^{\infty} \alpha_k \phi(t-k). \tag{5}$$

hint: remember, our goal is to show that the PWL interpolation is a basis function interpolation *with basis function ϕ. In that sense, the α_k have already been chosen (consult your notes and the relavent discussion handout), and you need only verify that they are correct.*

4 Lagrange Interpolation by Polynomials

Given n distinct points and the corresponding sampling of a function $f(x)$, $(x_i, f(x_i))$ for $0 \leq i \leq n-1$, the Lagrange polynomial interpolation is the polynomial of the least degree that passes through all of the given points.

Given n distinct points and the corresponding evaluations, $(x_i,\ f(x_i))$ for $0 \leq i \leq n-1$, the Lagrange polynomial interpolation is the $n-1^{th}$ degree polynomial

$$P(x) = \sum_{i=0}^{i=n-1} f(x_i) L_i(x),$$

where

$$L_i(x) = \prod_{j=0, j\neq i}^{j=n-1} \frac{(x-x_j)}{(x_i-x_j)} = \frac{(x-x_0)}{(x_i-x_0)} \cdots \frac{(x-x_{i-1})}{(x_i-x_{i-1})} \frac{(x-x_{i+1})}{(x_i-x_{i+1})} \cdots \frac{(x-x_{n-1})}{(x_i-x_{n-1})}. \tag{6}$$

Here is an example: for two data points, $(x_0, f(x_0))$ = (0, 4), $(x_1, f(x_1))$ = (-1, -3), we have

$$L_0(x) = \frac{x-x_1}{x_0-x_1} = \frac{x-(-1)}{0-(-1)} = x+1$$

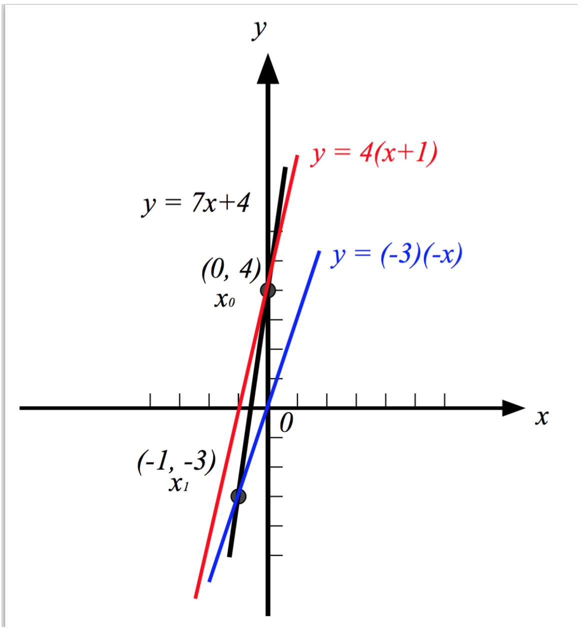
and

$$L_1(x) = \frac{x-x_0}{x_1-x_0} = \frac{x-(0)}{(-1)-(0)} = -x.$$

Then

$$P(x) = f(x_0)L_0(x) + f(x_1)L_1(x) = 4(x+1) + (-3)(-x) = 7x+4.$$

We can sketch those equations on the 2D plane as follows:



In the figure above, the red line is the 0^{th} interpolating polynomial L_0 weighted by the 0^{th} function values $f(x_0)$, $y = f(x_0)L_0 = 4(x+1)$. The blue line is the 1^{st} interpolating polynomial L_1 weighted by the 1^{st} function values $f(x_1)$, $y = f(x_1)L_1 = (-3)(-x) = 3x$. The black line is the interpolated signal, $P(x) = 7x + 4$.

a) Before we find the Lagrange interpolation, let us first use interpolation by global polynomials so we can verify our Lagrange interpolation results. Using the polynomial function basis $\{1, x, x^2, \cdots x^{n-1}\}$, the interpolation problem can be cast into finding the coefficients $a_0, a_1, a_2, \cdots, a_{n-1}$ of the function

$$g(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{n-1} x^{n-1}$$

such that $g(x_i) = f(x_i)$ for n samples of a function $(x_i, f(x_i))$ with $i \in \{0, 1, 2, ..., n-1\}$.

Given three data points, $(2,3)$, $(0,-1)$, and $(-1,-6)$, find a polynomial $g(x) = a_2 x^2 + a_1 x + a_0$ fitting the three points using global polynomial interpolation. Is this polynomial unique? That is, is it the only second degree polynomial that fits this data?

It is computationally expensive to do this process for large numbers of points, which is why we use the Lagrange interpolation method.

b) The set of Lagrange polynomials $\{L_i(x)\}$, $i \in \{0, 1, 2, ..., n-1\}$ is a new function basis for the subspace of degree $n-1$ or lower polynomials. **Find the $L_i(x)$ given by Eq. 6 corresponding to the three sample points in (a).** Show your work.

c) $P(x)$ is the sum of the Lagrange polynomials weighted by the function value at the corresponding points, giving the Lagrange interpolation of the given points. **Find the Lagrange polynomial interpolation $P(x)$ that goes through the three points in (a). Compare the result to the global polynomial interpolation of the same points, which you calculated in (a). Are they different from each other? Why or why not? Reason using the degree of the polynomials.**

d) **Plot $P(x)$ and each $f(x_i)L_i(x)$.** You can use a plotting utility (e.g. matplotlib) and or plot by hand.

e) **Show that** $P(x_i) = f(x_i)$ **for all** x_i**.** That is, show that the Lagrange interpolation passes through all given data points. Show this symbolically in the general case, not just for the example above.

5 Homework Process and Study Group

Citing sources and collaborators are an important part of life, including being a student!
We also want to understand what resources you find helpful and how much time homework is taking, so we can change things in the future if possible.

a) **What sources (if any) did you use as you worked through the homework?**

b) **If you worked with someone on this homework, who did you work with?**
List names and student ID's. (In case of homework party, you can also just describe the group.)

c) **Roughly how many total hours did you work on this homework?**

d) **Do you have any feedback on this homework assignment?**