EECS 16A Spring 2020 — Designing Information Devices and Systems I — Homework 14

This is an optional homework. Everything in this homework is in scope for the final, but you do not need to turn anything in. There are no self-grades for this homework.

1. OMP Exercise

Suppose we have a vector $\vec{x} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix}^T \in \mathbb{R}^4$. The vector $\vec{x}$ is related to the vector $\vec{b}$ in the following way:

$$\mathbf{M}\vec{x} \approx \vec{b}$$

$$\begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} & 0 \\ 0 & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \approx \begin{bmatrix} 4 \\ 6 \\ 3 \end{bmatrix}$$

For this undetermined and possibly noisy problem of finding $\vec{x}$, assume that $\vec{x}$ is sparse: it has 2 non-zero entries and 2 zero entries. Use Orthogonal Matching Pursuit to estimate x_1 to x_4.

(Note: Unlike previous examples you may have seen, we will not cross correlate $\vec{b}$ with the columns of $\mathbf{M}$. Instead, we will just compute the inner product of $\vec{b}$ with every column of $\mathbf{M}$.)

Solution:

Let $\vec{c}_1$ to $\vec{c}_4$ be the column vectors of $\mathbf{M}$. We first find the column vector in $\mathbf{M}$ that correlates most with $\vec{b}$:

$$\begin{aligned} \left\langle \vec{c}_1, \vec{b} \right\rangle &= 5\sqrt{2} \\ \left\langle \vec{c}_2, \vec{b} \right\rangle &= \frac{7\sqrt{2}}{2} \\ \left\langle \vec{c}_3, \vec{b} \right\rangle &= -\frac{3\sqrt{2}}{2} \\ \left\langle \vec{c}_4, \vec{b} \right\rangle &= -\frac{\sqrt{2}}{2} \end{aligned}$$

Thus, $\vec{c}_1$ is the best matching vector. Now we compute the error vector by computing the projection of $\vec{b}$ onto $\vec{c}_1$, and subtracting it from $\vec{b}$:

$$\vec{e} = \vec{b} - \left\langle \vec{b}, \vec{c}_1 \right\rangle \vec{c}_1 = \begin{bmatrix} -1 \\ 1 \\ 3 \end{bmatrix}.$$

Then we find the column that has the largest inner product with $\vec{e}$:

$$\begin{aligned} \langle \vec{c}_1, \vec{e} \rangle &= 0 \\ \langle \vec{c}_2, \vec{e} \rangle &= \sqrt{2} \\ \langle \vec{c}_3, \vec{e} \rangle &= \sqrt{2} \\ \langle \vec{c}_4, \vec{e} \rangle &= 2\sqrt{2} \end{aligned}$$

We know that $\vec{c}_1$ and $\vec{c}_4$ contribute most to $\vec{b}$, and now we project $\vec{b}$ onto the span of $\vec{c}_1$ and $\vec{c}_4$ using least squares. Let

$$\mathbf{A} = \begin{bmatrix} | & | \\ \vec{c}_1 & \vec{c}_4 \\ | & | \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & 0 \\ 0 & \frac{\sqrt{2}}{2} \end{bmatrix},$$

and the least squares formula gives:

$$\begin{bmatrix} x_1 \\ x_4 \end{bmatrix} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\vec{b} = \begin{bmatrix} \frac{19\sqrt{2}}{3} \\ \frac{8\sqrt{2}}{3} \end{bmatrix}.$$

Thus, $\vec{x} \approx \begin{bmatrix} \frac{19\sqrt{2}}{3} \\ 0 \\ 0 \\ \frac{8\sqrt{2}}{3} \end{bmatrix}$.

2. Greedy Algorithm for Calculating Matrix Eigenvalues

In the real world, it is not computationally practical to directly solve for the eigenvectors for large matrices as you might do for small matrices on paper. You would like to build an algorithm that sequentially computes the eigenvectors for a square symmetric matrix $\mathbf{Q} = \mathbf{A}^T\mathbf{A}$ (Note: A symmetric matrix $\mathbf{P}$ is a matrix such that its transpose is equal to itself, i.e. $\mathbf{P}^T = \mathbf{P}$. Component-wise this means $p_{ij} = p_{ji}$.)

To accomplish this we are given access to a function,

$$(\vec{v}_1, \lambda_1) = f(\mathbf{Q}), \tag{1}$$

that returns the **largest** eigenvalue of the matrix $\mathbf{Q}$ and the corresponding eigenvector. You do not need to know about the origins of the function, but if you are curious to learn more you can look up "Power iteration". We will use this function to build a greedy algorithm similar in the spirit of orthogonal matching pursuit that computes the eigenvectors and eigenvalues in descending order. By greedy algorithm we mean that we will choose eigenvectors one by one, at any time picking the eigenvector corresponding to the largest eigenvalue.

IMPORTANT: Throughout the problem, assume that the magnitude of each eigenvector is 1 (i.e. $\|\vec{v}_i\| = 1$), that the eigenvalues are real, unique, and distinct, and that the eigenvalues are indexed in a descending order $\lambda_1 > \lambda_2 > \cdots > \lambda_N > 0$.

(a) Let $\mathbf{A} \in \mathbb{R}^{M\times N}$. Let $\mathbf{Q} = \mathbf{A}^T\mathbf{A}$. Show that $\mathbf{Q} \in \mathbb{R}^{N\times N}$ is symmetric.
Hint: Recall that $(\mathbf{AB})^\mathbf{T} = \mathbf{B}^\mathbf{T}\mathbf{A}^\mathbf{T}$
Solution: Applying the hint, we have:

$$\mathbf{Q}^T = (\mathbf{A}^T\mathbf{A})^T = \mathbf{A}^T(\mathbf{A}^T)^T = \mathbf{A}^T\mathbf{A} = \mathbf{Q}$$

(b) Let us consider the matrix $\mathbf{V} \in \mathbb{R}^{N\times N}$:

$$\mathbf{V} = \begin{bmatrix} | & | & & | \\ \vec{v}_1 & \vec{v}_2 & \dots & \vec{v}_N \\ | & | & & | \end{bmatrix}. \tag{2}$$

Show that if $\langle \vec{v}_i, \vec{v}_j \rangle = 0$ for $i, j = \{1, \ldots, N\}$ when $i \neq j$ (all the columns of $\mathbf{V}$ are orthogonal to each other), then $\mathbf{V}^T\mathbf{V} = \mathbf{I}$.

Solution:

$$\mathbf{V}^T\mathbf{V} = \begin{bmatrix} \vec{v}_1^T\vec{v}_1 & \vec{v}_2^T\vec{v}_1 & \ldots & \vec{v}_n^T\vec{v}_1 \\ \vec{v}_2^T\vec{v}_1 & \vec{v}_2^T\vec{v}_2 & \ldots & \vec{v}_n^T\vec{v}_2 \\ \vdots & \vdots & & \vdots \end{bmatrix} = \mathbf{I} \tag{3}$$

All $\vec{v}_i^T\vec{v}_i = \|\vec{v}_i\|^2 = 1$ and when $i \neq j$ all $\vec{v}_i^T\vec{v}_j = 0$ because the eigenvectors are orthogonal.

(c) Show that if the columns of matrix $\mathbf{V}$ are orthogonal to each other, then the columns form a basis for $\mathbb{R}^N$.

Solution: We want to show that the columns of $\mathbf{V}$ form a basis for $\mathbb{R}^N$. To show that the columns form a basis for $\mathbb{R}^N$ we need to show two things:

- The columns must form a set of N linearly independent vectors.
- Any vector $\vec{x} \in \mathbb{R}^N$ can be represented as a linear combination of the vectors in the set.

We already know we have N vectors, so first we will show they are linearly independent. We shall do this by showing that $\mathbf{V}\vec{\beta} = \vec{0}$ implies that $\vec{\beta}$ can be only $\vec{0}$.

$$\mathbf{V}\vec{\beta} = \vec{0} \tag{4}$$

$$\beta_1\vec{v}_1 + \ldots + \beta_N\vec{v}_N = \vec{0} \tag{5}$$

Then to exploit the properties of orthogonal vectors, we consider taking the inner product of each side of the above equation with $\vec{v}_i$.

$$\langle \vec{v}_i, \beta_1\vec{v}_1 + \ldots + \beta_N\vec{v}_N \rangle = \langle \vec{v}_i, \vec{0} \rangle = 0 \tag{6}$$

Now we apply the distributive property of the inner product and the definition of orthonormal vectors,

$$\langle \vec{v}_i, \beta_1\vec{v}_1 \rangle + \ldots + \langle \vec{v}_i, \beta_i\vec{v}_i \rangle + \ldots + \langle \vec{v}_i, \beta_N\vec{v}_N \rangle = 0 \tag{7}$$

$$0 + \ldots + \beta_i\langle \vec{v}_i, \vec{v}_i \rangle + \ldots + 0 = 0 \tag{8}$$

$$0 + \ldots + \beta_i\vec{v}_i^T\vec{v}_i + \ldots + 0 = 0 \tag{9}$$

Because $\vec{v}_i^T\vec{v}_i = 1$, $\beta_i = 0$ for the equation to hold. Then, since this is true for all i from 1 to N, all the elements of the vector beta must be zero ($\vec{\beta} = \vec{0}$). Because $\vec{x} = \vec{0}$ implies $\vec{\beta} = \vec{0}$, the columns of $\mathbf{V}$ are linearly independent.

Now, we will show that any vector $\vec{x} \in \mathbb{R}^N$ can be represented as a linear combination of the columns of $\mathbf{V}$.

$$\vec{x} = \mathbf{V}\vec{\beta} = \beta_1\vec{v}_1 + \ldots + \beta_N\vec{v}_N \tag{10}$$

Because we know that the N columns of $\mathbf{V}$ are linearly independent, then there exists $\mathbf{V}^{-1}$. Applying the inverse to the equation above,

$$\mathbf{V}^{-1}\mathbf{V}\vec{\beta} = \mathbf{V}^{-1}\vec{x} \tag{11}$$

$$\vec{\beta} = \mathbf{V}^{-1}\vec{x}, \tag{12}$$

we find that there exists a unique $\vec{\beta}$ that allow us to represent any $\vec{x}$ as a linear combination of the columns of $\mathbf{V}$.

(d) Assume that the columns of matrix $\mathbf{V}$ are orthogonal to each other for the remainder of the problem. Since the columns of $\mathbf{V}$ form a basis, let $\vec{b} = \alpha_1\vec{v}_1 + \cdots + \alpha_N\vec{v}_N$ for some scalars α_i. Find $\langle \vec{v}_i, \vec{b} \rangle$.

Solution:

$$\langle \vec{v}_i, \vec{b} \rangle = \vec{v}_i^T\vec{b} = \vec{v}_i^T \sum_{j=1}^{N} \alpha_j\vec{v}_j = 0 + \cdots + \alpha_i\vec{v}_i^T\vec{v}_i + \cdots + 0 = \alpha_i \tag{13}$$

(e) Find the $\hat{\vec{x}}$ that minimizes the following least squares problem:

$$\min_{\vec{x}} ||\mathbf{V}\vec{x} - \vec{b}||$$

Let $\vec{e} = \mathbf{V}\vec{x} - \vec{b}$. Also find $||\vec{e}||$ for the optimal $\hat{\vec{x}}$.

Hint: Use what you proved in the earlier parts of this problem.

Solution: We can apply the least squares formula to find:

$$\begin{aligned}\hat{\vec{x}} &= (\mathbf{V}^T\mathbf{V})^{-1}\mathbf{V}^T\vec{b} \\ &= \mathbf{I}\mathbf{V}^T\vec{b} \\ &= \mathbf{V}^T\vec{b}\end{aligned}$$

$$\begin{aligned}||\vec{e}|| &= ||\mathbf{V}\hat{\vec{x}} - \vec{b}|| \\ &= ||\mathbf{V}\mathbf{V}^T\vec{b} - \vec{b}|| \\ &= ||\vec{0}|| \\ &= 0\end{aligned}$$

We know that $\vec{b}$ is in the column space of $\mathbf{V}$ and that the norm of any vector is greater than or equal to zero. Therefore, the best possible solution to the least squares problem is zero. We are able to achieve zero when we pick $\hat{\vec{x}}$ such that $\mathbf{V}\hat{\vec{x}} = \vec{b}$, and this is possible because $\vec{b}$ is in the column space of $\mathbf{V}$.

(f) Let us define

$$\mathbf{V}_2 = \begin{bmatrix} | & & | \\ \vec{v}_2 & \cdots & \vec{v}_N \\ | & & | \end{bmatrix} \tag{14}$$

where we have removed the first column of $\mathbf{V}$. Note that $\mathbf{V}_2 \in \mathbb{R}^{N\times(N-1)}$, and $\mathbf{V}_2$ is not invertible. Assume $\vec{\alpha} = [\alpha_1 \dots \alpha_N]^T$ are the same weights as in part (d). Find the $\hat{\vec{x}}$ that minimizes the following least squares problem:

$$\min_{\vec{x}} ||\mathbf{V}_2\vec{x} - \vec{b}||.$$

Let $\vec{e} = \mathbf{V}_2\vec{x} - \vec{b}$. Also find $||\vec{e}||$.

Solution: We apply the least squares formula to find:

$$\begin{aligned}
\hat{\vec{x}} &= (\mathbf{V}_2^T\mathbf{V}_2)^{-1}\mathbf{V}_2^T\vec{b} \\
&= \mathbf{I}\mathbf{V}_2^T\vec{b} \\
&= \mathbf{V}_2^T\mathbf{V}_1\vec{\alpha} \\
&= \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N \end{bmatrix} \\
&= \begin{bmatrix} \alpha_2 \\ \alpha_3 \\ \vdots \\ \alpha_N \end{bmatrix}
\end{aligned}$$

$$\begin{aligned}
\mathbf{V}_2\hat{\vec{x}} &= \begin{bmatrix} | & & | \\ \vec{v}_2 & \dots & \vec{v}_N \\ | & & | \end{bmatrix} \begin{bmatrix} \alpha_2 \\ \alpha_3 \\ \vdots \\ \alpha_N \end{bmatrix} \\
&= (\alpha_2\vec{v}_2 + \alpha_3\vec{v}_3 + \cdots + \alpha_N\vec{v}_N)
\end{aligned}$$

$$\begin{aligned}
||\vec{e}|| &= ||\mathbf{V}_2\hat{\vec{x}} - \vec{b}|| \\
&= ||(\alpha_2\vec{v}_2 + \alpha_3\vec{v}_3 + \cdots + \alpha_N\vec{v}_N) - (\alpha_1\vec{v}_1 + \alpha_2\vec{v}_2 + \cdots + \alpha_N\vec{v}_N)|| \\
&= ||-\alpha_1\vec{v}_1|| = |\alpha_1|
\end{aligned}$$

(g) Consider a matrix $\mathbf{Q}$ given as:

$$\mathbf{Q} = \mathbf{V}\Lambda\mathbf{V}^T = \sum_{i=1}^{N} \lambda_i \vec{v}_i \vec{v}_i^T \tag{15}$$

$$\text{where } \Lambda = \begin{bmatrix} \lambda_1 & 0 & 0 & \dots & 0 \\ 0 & \lambda_2 & 0 & \dots & 0 \\ 0 & 0 & \lambda_3 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \lambda_N \end{bmatrix} \tag{16}$$

It turns out that the eigenvectors of $\mathbf{Q}$ are given by the columns of $\mathbf{V}$. Try to prove this yourself.

Consider:

$$\begin{aligned}\mathbf{Q}\vec{v}_1 &= \mathbf{V}\Lambda\mathbf{V}^T\vec{v}_1 \\ &= \mathbf{V}\Lambda\begin{bmatrix}1\\0\\\vdots\\0\end{bmatrix} \\ &= \mathbf{V}\begin{bmatrix}\lambda_1\\0\\\vdots\\0\end{bmatrix} \\ &= \lambda_1\vec{v}_1\end{aligned}$$

Can you justify why each of the steps in the proof above is correct?

Solution:

$\mathbf{Q}\vec{v}_1 = \mathbf{V}\Lambda\mathbf{V}^T\vec{v}_1$

$= \mathbf{V}\Lambda\begin{bmatrix}1\\0\\\vdots\\0\end{bmatrix}$ (The vector $\vec{v}_1$ is orthogonal to all other vectors $\vec{v}_i$ for $i \neq 1$, and $\vec{v}_1^T v_1 = 1$.)

$= \mathbf{V}\begin{bmatrix}\lambda_1\\0\\\vdots\\0\end{bmatrix}$ (The first column of Λ is extracted as a result from the matrix multiplication.)

$= \lambda_1\vec{v}_1$ (The first column of $\mathbf{V}$ multiplied by λ_1 remains as a result from the matrix multiplication.)

(h) Now consider $\mathbf{Q}_2 = \mathbf{Q} - \lambda_1\vec{v}_1\vec{v}_1^T$. Thus, $\mathbf{Q}_2$ represents $\mathbf{Q}$ after the component associated with direction $\vec{v}_1$ is removed. Show that $\vec{v}_1$ is in the null space of $\mathbf{Q}_2$, and $\vec{v}_2$ to $\vec{v}_N$ are eigenvectors of $\mathbf{Q}_2$.

Hint: *Can you write* $\mathbf{Q}_2$ *using Eq. (15)?*

Solution: To show that $\vec{v}_1$ is in the nullspace of $\mathbf{Q}_2$, we want to show that $\mathbf{Q}_2\vec{v}_1 = \vec{0}$.

$$\begin{aligned}\mathbf{Q}_2 &= \mathbf{Q} - \lambda_1\vec{v}_1\vec{v}_1^T \\ &= \sum_{i=1}^{N}\lambda_i\vec{v}_i\vec{v}_i^T - \lambda_1\vec{v}_1\vec{v}_1^T \\ &= \sum_{i=2}^{N}\lambda_i\vec{v}_i\vec{v}_i^T \\ \mathbf{Q}_2\vec{v}_1 &= \sum_{i=2}^{N}\lambda_i\vec{v}_i\vec{v}_i^T\vec{v}_1\end{aligned}$$

Since $\vec{v}_1$ is orthogonal to all of the other eigenvectors, $\vec{v}_i^T \vec{v}_1 = 0$ for all $i \neq 1$.

$$\begin{aligned}\mathbf{Q}_2 \vec{v}_1 &= \sum_{i=2}^{N} \lambda_i \vec{v}_i 0 \\ &= \vec{0}\end{aligned}$$

We've shown that $\vec{v}_1$ is in the null space of $\mathbf{Q}_2$.

Now we show that $\vec{v}_2$ to $\vec{v}_N$ are eigenvectors of $\mathbf{Q}_2$.

$$\mathbf{Q}_2 \vec{v}_j = \sum_{i=2}^{N} \lambda_i \vec{v}_i \vec{v}_i^T \vec{v}_j \qquad \text{for} j = 2, \ldots, N$$

Since $\vec{v}_j$ is orthogonal to all of the other eigenvectors, $\vec{v}_i^T \vec{v}_j = 0$ for all $i \neq j$. Also, $\vec{v}_i^T \vec{v}_j = 1$ for $i = j$. The above expression becomes:

$$\mathbf{Q}_2 \vec{v}_j = \lambda_j \vec{v}_j \qquad \text{for} j = 2, \ldots, N$$

We have shown that $\vec{v}_2$ to $\vec{v}_N$ are eigenvectors of $\mathbf{Q}_2$

(i) Recall the function that returns the **largest** eigenvalue and corresponding eigenvector of a matrix,

$$(\vec{v}_1, \lambda_1) = f(\mathbf{Q}). \tag{17}$$

Design an algorithm that returns a list of eigenvalues of matrix $\mathbf{Q}$ in descending order of values. You may assume that all the eigenvalues of $\mathbf{Q}$ are positive (> 0). You are allowed to use the function defined in Eq. (17) that returns the largest eigenvalue and corresponding eigenvector and what you know from part (g).

Solution:

```
function OMP(Q):
    list_of_eigenvalues = []
    For i in range(0,N):
        (v, λ) = f(Q)
        list_of_eigenvalues.append(λ)
        Q = Q − λvv^T
    return list_of_eigenvalues
```

What is this algorithm above doing? It is using the function that returns the largest eigenvalue (and corresponding eigenvector) and "peeling off" the eigenvectors one by one. After we get an eigenvector, we remove it from the matrix by $\mathbf{Q} - \lambda \vec{v}\vec{v}^T$. The remainder of the eigenvectors are preserved and in the next iteration of the loop, the eigenvector corresponding to the next largest eigenvector is picked and returned.

3. Sparse Imaging

Recall the imaging lab where we projected masks onto an object to scan it into our computer using a single pixel measurement device, that is, a photoresistor. In that lab, we were scanning a 30×40 image having 1200 pixels. In order to recover the image, we took exactly 1200 measurements because we wanted our

'mask matrix' $\mathbf{A}$ to be invertible. In this problem, we have a 2D image $\mathbf{I}$ of size 91×120 pixels for a total of 10920 pixels. We will explore how to reconstruct the image using only 6500 measurements.

The image is made up of mostly black pixels (represented by a zero) except for 476 white ones (represented by a one). In cases where there are a small number of non-black pixels, we can reduce the overall number of samples necessary using the orthogonal matching pursuit algorithm. This reduces the time required for scanning an image, a real-world concern for lengthy processes like MRI where people have to stay still while being imaged.

Although the imaging illumination masks we used in the lab consisted of zeros and ones, in this question, we are going to have masks with real values. Say that we have an imaging mask $\mathbf{M_0}$ of size 91×120. The measurements using this imaging mask can be represented as follows:

First, let us put every element in the matrix $\mathbf{I}$ into a column vector $\vec{i}$ of length 10920. This operation is referred to as vectorization. Likewise, let us vectorize the mask $\mathbf{M_0}$ to $\vec{m}_0$ which is a column vector of length 10920. Then the measurement using the mask $\mathbf{M_0}$ can be represented as

$$b_0 = \vec{m}_0^T \vec{i}.$$

Say we have a total of K measurements, each taken with a different illumination mask. Then, these measurements can collectively be represented as

$$\vec{b} = \mathbf{A}\vec{i},$$

where $\mathbf{A}$ is an $K \times 10920$ size matrix whose rows contain the vectorized forms of the illumination masks, that is

$$\mathbf{A} = \begin{bmatrix} — & \vec{m}_1^T & — \\ — & \vec{m}_2^T & — \\ & \vdots & \\ — & \vec{m}_K^T & — \end{bmatrix}.$$

To show that we can reduce the number of samples necessary to recover the sparse image $\mathbf{I}$, we are going to only generate 6500 masks. We will generate $\mathbf{A}$ so that the columns of $\mathbf{A}$ are approximately orthogonal with each other. The following question refers to the part of IPython notebook file accompanying this homework related to this question.

(a) In the IPython notebook, we call a function `simulate` that generates masks and the measurements. You can see the masks and the measurements in the IPython notebook file. Complete the function `OMP` that does the OMP algorithm described in lecture.

Solution:

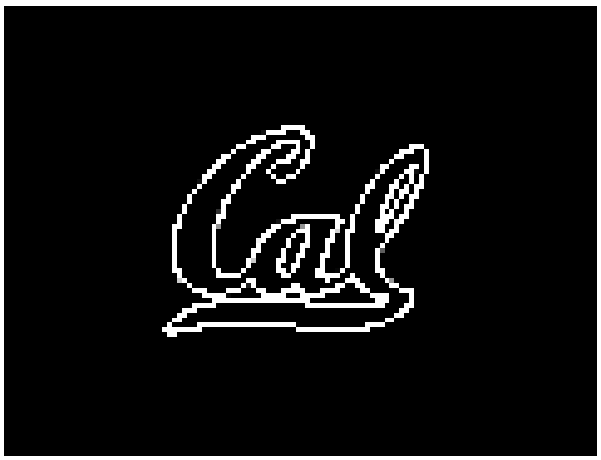
See `sol14.ipynb`.

Remark: When you look at the vector `measurements` you will see that it has zero average value. Likewise, the columns of the matrix containing the masks $\mathbf{A}$ also have zero average value. To satisfy these conditions, they need to have negative values. However, in an imaging system, we cannot project negative light. One way to get around this problem is to find the smallest value of the matrix $\mathbf{A}$ and subtract it from all entries of $\mathbf{A}$ to get the actual illumination masks. This will yield masks with positive values, hence we can project them using our real-world projector. After obtaining the readings using these masks, we can remove their average value from the readings to get measurements as if we had multiplied the image using the matrix $\mathbf{A}$.

(b) Run the code `rec = OMP((height, width), sparsity, measurements, A)` and see the image being correctly reconstructed from a number of samples smaller than the number of pixels of your figure. What is the image?

Solution:

The reconstructed image is the following.



(c) **PRACTICE:** We have supplied code that reads a PNG file containing a sparse image, takes measurements, and performs OMP to reconstruct the original image. An example input image file is also supplied together with the code. Recover `smiley.png` using OMP with 6500 measurements.

You can answer the following parts of this question in very general terms. Try reducing the number of measurements. Does the algorithm start to fail in recovering your sparse image? Why do you think it fails?

Solution:

The answer to this question depends on the size of your input image and the total number of non-zero pixels in it. In order to successfully recover the image, the number of measurements (masks used for imaging) needs to increase as the number of non-zero pixels increases. The reason is that as we have more nonzero pixels, they start to contribute to our measurements. In order to better distinguish these contributions, we need to take more measurements with different masks.