*Chapter 11*

# Day 6: AI Discussion, Smile Detection and Eigenthings

## 11.1 Schedule

- 0900-0915: Debrief
- 0915-1000: AI and Society Discussion
- 1000-1030: Smile Detection—Concepts
- 1030-1045: Coffee
- 1045-1115: Machine Learning
- 1115-1145: Smile Detection—Implementation
- 1145-1210: Eigenthings
- 1210-1230: Review/preview

## 11.2 Debrief and Synthesis [15 mins]

- Please discuss your overnight work with your table-mates, and get help with the ideas that you are still confused by.

## 11.3 AI and Society Discussion [45 mins]

### Framing (5 minutes)

Today we'll be talking about a constellation of issues that arise when AI technology, like facial recognition, is deployed in society. As the historian Melvin Kranzberg famously remarked, "Technology is neither good nor bad; nor is it neutral." As you saw in the reading from the night assignment, the effect of AI technology in society intersects a number of sensitive issues around race, class, and gender. Due to intersection of AI and these sensitive issues, it helps to take a few minutes to consider some guidelines for having fruitful discussions at your tables.

- Check out this poster put together by some Oliners with suggestions for having conversations on sensitive topics.
- The readings provide common information and framing, which we find is very helpful to finding common ground when discussing issues that individuals may relate to in very different ways.
- As you may be relatively new to these ideas, consider adopting a mindset of identifying key questions rather than necessarily coming to conclusions.
- When talking about the effect of a technology on a group that has been historically oppressed, you should be particularly sensitive in these discussions if you are not a member of this group. Be conscious of the ways in which your words might be experienced by those who may have faced a history of discrimination due to being a member of this group.

### *Unpacking the Readings (15 minutes)*

Write down key concepts and clear up points of confusion on the readings.

- Joy Buolamwini's written testimony on bias in facial recognition technology (you may have watched this instead).
- Principles for Accountable Algorithms
- Google's Inclusive ML

### *Share Your Positive Application of AI (10 minutes)*

Go around and share the application of AI that you think has the potential for great positive impact on society. Say a little bit about what you learned and how you think it would have a positive impact (e.g., in what ways and for whom).

### *What Did You Take Away? (15 minutes)*

**Please have one person take notes on this in some electronic format so they can submit it as part of their day survey**

As a table, discuss what you took away from the readings and your discussion thus far. Here are some dimensions that you might want to explore.

1. What parts or quotes from the readings were most surprising / impactful to you?
2. Were you surprised by your reaction to reading any of the material (e.g., felt unexpectedly angry, sad, indifferent)?
3. What are the big questions that have been raised for you (these could be things that were already on your radar or new ones entirely)? These questions could relate to our society as a whole, your role as a citizen within society, your role as an Olin student, your future career path, etc.).
4. How do these readings intersect with knowledge you've gained from other contexts (e.g., in other courses or in your daily life experience)?

## *11.4 Smile Detection—Concepts [30 mins]*

In this session we are going to use our toolbox of linear algebra skills to "detect" whether or not a person is smiling in a photograph. The approach that we will take is very common in **machine learning** - we will use a dataset to **train** our algorithm, and we will use a different dataset to **test** our algorithm. We will first develop the conceptual framework and then implement the approach in MATLAB.

### *The Big Idea*

Let's assume that we have 100 training photos of faces, each consisting of a 5 by 5 grid of pixels. Let's pack these into a matrix $\mathbf{A}$ with 100 rows and 25 columns, i.e. every row is a different face and every column is a different pixel.

Let's also assume that we have already classified every training face as "smiling" or "not-smiling". Let's create a column vector $\mathbf{b}$ with 100 rows (corresponding to each face) which has either $1$ (smiling) or $0$ (not-smiling).

Let's now develop a linear system of algebraic equations by trying to express the vector $\mathbf{b}$ as a linear combination of the columns of $\mathbf{A}$, i.e.

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

Notice that the vector $\mathbf{x}$ is a column vector with 25 rows - one row for each pixel. Since there are more rows than columns we know that an **exact** solution does not exist, so we will find the **approximate** solution by orthogonal projection, i.e. we will solve

$$\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{b}$$

Now that we have the vector $\mathbf{x}$, let's use it to detect whether a test image is smiling. Assuming that the test image is packed into a single row vector $\mathbf{t}$ (with 25 columns) then the product

$$\mathbf{t}\mathbf{x}$$

will return a scalar. If this scalar is close to "1" then we predict the face is smiling. If this scalar is close to "0" then we predict the face is not smiling.

**Exercise 11.1**

In this exercise you will be carefully reading and interpreting this big idea. We are including these questions as a scaffold, pointing out interesting features along the way.

1. Read "The Big Idea" again!
2. Interpret what it means to write down the linear system of equations $\mathbf{A}\mathbf{x} = \mathbf{b}$ and give a meaning to the vector $\mathbf{x}$.
3. Interpret the product $\mathbf{A}^T\mathbf{A}$ and the product $\mathbf{A}^T\mathbf{b}$.
4. The vector $\mathbf{x}$ does not satisfy $\mathbf{A}\mathbf{x} = \mathbf{b}$ exactly. What does the expression $\mathbf{A}\mathbf{x} - \mathbf{b}$ tell you?
5. How would you decide whether your "trained" algorithm was worth using on a test dataset?
6. Assume you had 40 test images with 25 pixels each and that you pack them into a matrix $\mathbf{T}$ with 40 rows and 25 columns. Write down the matrix-vector product you would use for smile detection on this test dataset.
7. How would you measure the accuracy of your predictions if we also provided you with the data on whether each test image was smiling or not?

## 11.5 Machine Learning in general [30 mins]

Machine learning is an interdisciplinary field concerned with the idea that rather than preprogramming machines to solve tasks explicitly, instead we can program them to learn to solve tasks through experience. In order to connect this idea to what you've done thus far, first we'll introduce a somewhat more formal definition of machine learning.

> *A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E.*
>
> — *Tom Mitchell*

Let's take the smile detection problem that you just framed in the preceding section of the document.

| Symbol from Mitchell's Definition | Smile Detection |
|---|---|
| $E$ | 5 by 5 grids of pixels with corresponding labels as to whether or not the person in the grid is smiling |
| $T$ | Given a new 5 by 5 grid of pixels, predicting whether the person in the image is smiling. |
| $P$ | Accuracy on predicting whether a person is smiling (e.g., percent correct) |

The last part of the definition states that in order to say a program is "learning", it should "improve with experience E." In the case of smile detection, what this means is that given more images with corresponding indication of whether each person is smiling, our program should get better at predicting whether a face is smiling. Later in this document when you implement smile detection, you'll see that the framing of smile detection as an LSAE problem absolutely meets this definition (and does a surprisingly good job at it too)!

**Exercise 11.2**

In this exercise you and your table-mates will be working to frame various machine learning problems as systems of linear equations. In doing so, you should answer the following questions.

- What is the thing you want to predict (i.e., what is your $\mathbf{b}$ in $\mathbf{Ax} = \mathbf{b}$).
- What quantities are you using to make your prediction (i.e., what do each of the rows and columns in the matrix $\mathbf{A}$ represent)?
- While $\mathbf{x}$ would be determined by solving $\mathbf{Ax} = \mathbf{b}$, come up with a guess as to what $\mathbf{x}$ would be (don't worry about coming up with numbers. Instead, try to identify the sign of each element of $\mathbf{x}$ and whether its magnitude is large or small relative to the other entries of $\mathbf{x}$).
- How would you measure how well your system works? For example, for smile detection we might apply our learned model $\mathbf{x}$ to new data and see how often it correctly predicts the facial expression (smiling vs. not smiling) of the person in each image.

- What sorts of issues of bias might you have to worry about in this system?

1. AirBnB has a smart pricing option that lets folks who list their properties on the site have the price for those listings determined automatically. How might you frame the creation of this smart pricing tool as solving a system of linear equations? We suggest that you follow the steps outlined at the start of this exercise. You might want to do a quick AirBnB search if you are not familiar with the site.

2. Netflix suggests content to a user that they are likely to enjoy based on their viewing history as well as the viewing histories of others. In the past (although we think possibly not anymore), they used to also incorporate ratings data (1 to 5 stars) of particular movies from both the user receiving the recommendation and other users on the site. How might you frame the creation of this recommendation system in terms of solving a system of linear equations? We suggest that you follow the steps outlined at the start of this exercise. You might want to do a quick Netflix search if you are not familiar with the site.

## *11.6 Smile Detection—Implementation [30 mins]*

Please download the file `smiles.mat` from the canvas site. If you load this file in MATLAB, you will then have access to the following variables in your workspace.

```
train_data       - a 3D array containing 19685 24 x 24 pixel
                   images of faces
smile_flag_train - a vector of the same length as the number
                   of images in train_data, with 1s indicating
                   which images are smiling
test_data        - 500 24 x 24 pixel images of faces
smile_flag_test  - a vector of the same length as the number
                   of images in test_data, with 1s indicating
                   which images are smiling
```

The 'train data' and the associated 'smile flag train' are the sets of data you should use to develop your mathematical model. The 'test data' and its associated 'smile flag test' are the sets of data you should use to test your algorithm when you are finished!

**Exercise 11.3**

For this exercise we recommend that you use our walkthrough notebook. The notebook has embedded solutions or you can try it with minimal scaffolding using the suggested process below. Even if you decide not to use the walkthrough notebook, it's worth running the embedded solutions to pickup some techniques for visualizing your smile detector model. In any case, as you move into actually implementing the smile detector, we recommend you work as pairs at your table. Working in a pair will allow you to have someone to bounce ideas off of and also make sure you can see the

laptop easily.
Only if you decide NOT to use the walkthrough notebook, you should consider following the procedure below to implement the smile detector.

1. Sketch out a set of steps you would take in order to implement smile detection. (Just words here - no code. e.g. we will have to pack all images into a single matrix)
2. Turn this set of steps into MATLAB pseudo-code. Identify important coding elements without implementing, e.g. we will use **reshape** to pack the given dataset into a matrix.
3. Review the documentation for MATLAB functions that will be used and be clear on how to use them before implementation, e.g. » help reshape
4. Methodically implement smile detection in MATLAB, testing as you go.

## 11.7 Introduction to Eigenthings [25 mins]

We are now going to learn the secret of the genie ...

### Eigenvalues and Eigenvectors: Definition and Notation

Consider a square $n \times n$ matrix $\mathbf{A}$. A vector $\mathbf{v}$ is said to be an eigenvector of $\mathbf{A}$ with corresponding eigenvalue $\lambda$ if $\mathbf{v}$ is not a vector of all zeros, and

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v}. \tag{11.1}$$

If we treat $\mathbf{A}$ as a transformation matrix then $\mathbf{v}$ is an eigenvector of $\mathbf{A}$ if it is simply scaled when acted on by the matrix $\mathbf{A}$. In other words, $\mathbf{v}$ does not change direction when acted upon by $\mathbf{A}$. In general, an $n \times n$ matrix has exactly $n$ eigenvalues (although some of these may be repeated and some of these may be complex!). Note that any scalar multiple of an eigenvector of a matrix is also an eigenvector of that matrix - it's only the direction of the eigenvector that matters.

In the next overnight assignment we are going to develop formal techniques for finding the eigenvalues and eigenvectors of matrices. For now, we are going to focus on concepts and developing some intuition.

**Exercise 11.4**

1. Show that $\mathbf{v} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$ is an eigenvector of the following matrix by computing the product $\mathbf{A}\mathbf{v}$, and find the corresponding eigenvalue

$$A = \begin{bmatrix} 1 & 1 \\ 2 & 0 \end{bmatrix} \tag{11.2}$$

2. On the same axes, plot the vector representing $\mathbf{v} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$ and $\mathbf{A}\mathbf{v}$. Does the plot confirm that this is an eigenvector?
3. On the same axes, plot the vector representing $\mathbf{u} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 1 \end{bmatrix}$ and $\mathbf{A}\mathbf{u}$. Is this is an eigenvector of $\mathbf{A}$?

*Eigenvalues and eigenvectors of a diagonal matrix*

Recall from our earlier work that the matrix

$$\mathbf{A} = \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}$$

scales vectors by a factor of 2 in the x-direction and by a factor of 3 in the y-direction. Thus a vector that had a non-zero component only in the $x$ direction will be scaled by a factor of 2 when transformed by this matrix. In other words, $\lambda_1 = 2$ is an eigenvalue with corresponding eigenvector $\mathbf{v}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, and that $\lambda_2 = 3$ is an eigenvalue with corresponding eigenvector $\mathbf{v}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$. Let's check if the first one is true:

$$\mathbf{A}\mathbf{v}_1 = \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \end{bmatrix} = 2 \begin{bmatrix} 1 \\ 0 \end{bmatrix} = 2\mathbf{v}_1$$

Therefore $\lambda_1 = 2$ is an eigenvalue with corresponding eigenvector $\mathbf{v}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$.

**Exercise 11.5**

Confirm that $\lambda_2 = 3$ is an eigenvalue with corresponding eigenvector $\mathbf{v}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ by computing the product $\mathbf{A}\mathbf{v}_2$.

Based on this example, we can heuristically guess that the eigenvalues of an $n \times n$ diagonal matrix are the entries on the diagonal. The $n$ eigenvectors each have a single 1 in them, with the remaining entries being zero.

**Exercise 11.6**

What are the eigenvalues and eigenvectors of the following diagonal matrices

1. $$\mathbf{A} = \begin{bmatrix} -3 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 4 \end{bmatrix}$$

2. $$\mathbf{A} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

**Exercise 11.7**

1. Assume that $\mathbf{v} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ is an eigenvector with eigenvalue 3. Construct an appropriate matrix $A$ with this eigenvalue and eigenvector by first rotating $\mathbf{v}$ onto the x-axis, scaling it by 3, and then rotating back.

**Exercise 11.8**

What is one eigenvector of the following matrix?

$$\mathbf{R} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \tag{11.6}$$

## 11.8 Review and Preview [20 minutes]

## Solution 11.1

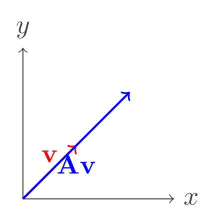
1. Read, read, read ....
2. We are trying to take a linear combination of the data in order to predict whether each image is smiling or not. The vector $\mathbf{x}$ is the magic set of weights we have to use. Its size is the same as the number of pixels, so maybe it should look like a mask that we can place over an image to tell us whether it is smiling.
3. The product $\mathbf{A}^T\mathbf{A}$ is like a pixel to pixel correlation matrix, except we haven't scaled the data matrix $\mathbf{A}$. The product $\mathbf{A}^T\mathbf{b}$ is the sum of the images that are smiling.
4. The expression $\mathbf{Ax} - \mathbf{b}$ tells us the error in predicting whether a training image is smiling or not.
5. We could add up how often the predictor is correct and divide by the number of images to get an estimate of the accuracy. We would decide on a cut-off before we used it on a test dataset.
6. It is simply $\mathbf{Tx}$.
7. As before. Determine how many we got correct and average it.

## Solution 11.3

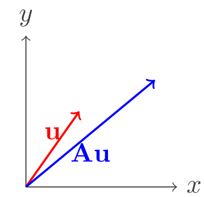
You can use the solutions that are embedded in the walkthrough notebook.

## Solution 11.4

1. Compute that
$$A\mathbf{v} = \begin{bmatrix} \sqrt{2} \\ \sqrt{2} \end{bmatrix} = 2\mathbf{v}$$
and so the corresponding eigenvalue is $\lambda = 2$.
2. As we can see in the picture below, both $\mathbf{v}$ and $\mathbf{Av}$ point in the same direction, which confirms $\mathbf{v}$ is an eigenvalue of $\mathbf{A}$




3. As we can see in the picture below, $\mathbf{u}$ and $\mathbf{Au}$ point in different directions, so $\mathbf{u}$ is not an eigenvector of $\mathbf{A}$




## Solution 11.5

We compute that

$$\mathbf{A}\mathbf{v}_2 = \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \end{bmatrix} = 3 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 3\mathbf{v}_2.$$

## Solution 11.6

1. The eigenvalues are $\lambda_1 = -3$, $\lambda_2 = -1$ and $\lambda_3 = 4$ and the corresponding eigenvectors are $\mathbf{v}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$, $\mathbf{v}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$ and $\mathbf{v}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$.
2. The eigenvalues are $\lambda_1 = 2$, $\lambda_2 = 4$ and $\lambda_3 = 0$ and the corresponding eigenvectors are $\mathbf{v}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$, $\mathbf{v}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$ and $\mathbf{v}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$.

## Solution 11.7

1. First rotate it using a 45 degree clockwise rotation matrix

$$\mathbf{R}(-45) = \begin{bmatrix} cosd(-45) & -sind(-45) \\ sind(-45) & cosd(-45) \end{bmatrix} \tag{11.3}$$

Now that it is along the x-axis we can scale it by 3 using the scaling matrix

$$S = \begin{bmatrix} 3 & 0 \\ 0 & 0 \end{bmatrix} \tag{11.4}$$

We then rotate it back using $\mathbf{R}(45)$. Multiplying these matrices together gives

$$\mathbf{A} = \begin{bmatrix} 1.5 & 1.5 \\ 1.5 & 1.5 \end{bmatrix} \tag{11.5}$$

## Solution 11.8

The vector $\mathbf{v} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$ is an eigenvector of $\mathbf{R}$ because it is the rotation axis and therefore remains unchanged on rotation.