EE 127 / EE 227AT 9/20/18
L. El Ghaoui

HW #2 Solutions

Exercise 1 (PCA and low-rank compression) We are given a $n \times m$ matrix $X = [x_1, \ldots, x_m]$, with $x_i \in \mathbb{R}^n$, $i = 1, \ldots, m$ the data points. We assume that the data matrix is centered, in the sense that $x_1 + \ldots + x_m = 0$. In lecture, it was asserted that there is equivalence between three problems:

(P_1) Finding a line going through the origin that maximizes the variance of the points projected on the line.

(P_2) Finding a line going through the origin that minimizes the sum of squares of the distances from the points to their projections;

(P_3) Finding a rank-one approximation to the data matrix.

In this exercise, you are asked to show the equivalence between these three problems.

1. Consider the problem of projecting a point x on a line $\mathcal{L} = \{x_0 + vu \ : \ v \in \mathbb{R}\}$, with $x_0, \in \mathbb{R}^n$, $u^T u = 1$, given. Show that the projected point z is given by $z = x_0 + v^* u$, with $v^* = (x - x_0)^T u$, and that the minimal squared distance $\|z - x\|_2^2$ is equal to $\|x - x_0\|_2^2 - ((x - x_0)^T u)^2$.
2. Show that problems P_1, P_2 are equivalent.
3. Show that P_3 is equivalent to P_1. *Hint:* show that the data matrix is rank-one if and only if the points are all on a line that goes through the origin.

Solution 1

1. The projection of point x on $\mathcal{L}$ corresponds to the following problem:

$$v^* = \min_v \ \|x_0 + vu - x\|_2.$$

The squared objective writes

$$\|x_0 + vu - x\|_2^2 = v^2 - 2v(x - x_0)^T u + \|x - x_0\|_2^2,$$

which proves that the optimal value of v is

$$v^* = (x - x_0)^T u.$$

At optimum, the squared objective function, which equals the minimum squared distance $\|z - x\|_2^2$, takes the desired value:

$$\|x_0 + v^* u - x\|_2^2 = \|x - x_0\|_2^2 - ((x - x_0)^T u)^2.$$

2. P_2 minimizes the sum of squared distances from the points to their projections on a line passing through the origin $\mathcal{L} = \{vu : v \in \mathbb{R}\}$. This problem can be written as:

$$\min_{u \,:\, u^T u = 1} \sum_{i=1}^m \min_{v_i} \|x_i - v_i u\|_2^2 \tag{1}$$

where $v_i = x_i^T u$, as defined in the previous part (note that $x_0 = 0$ because $\mathcal{L}$ passes through the origin). From the previous part, we obtain the equivalent form:

$$\begin{aligned} & \min_{u \,:\, u^T u = 1} \sum_{i=1}^m \|x_i\|_2^2 - (x_i^T u)^2 \\ = & \min_{u \,:\, u^T u = 1} \sum_{i=1}^m -(u^T x_i)(x_i^T u) \\ = & \max_{u \,:\, u^T u = 1} \sum_{i=1}^m u^T x_i x_i^T u \end{aligned}$$

This problem can be written as a variance maximization problem:

$$\max_{u \,:\, u^T u = 1} u^T C u,$$

where $C := (1/m) \sum_{i=1}^m x_i x_i^T$ is the covariance matrix associated with the centered data. The above is exactly P_1, which maximizes the variance of points projected on a line.

3. Assume that all the points are on a line going through the origin: this means that there exist $u \in \mathbb{R}^n$, with $u^T u = 1$, such that, for every i, there exist a scalar v_i such that

$$x_i = v_i u.$$

This means that the data matrix is rank-one:

$$X = [x_1, \ldots, x_m] = [v_1 u, \ldots, v_m u] = u v^T,$$

with $v^T = [v_1, \ldots, v_m]$.

Now P_3 is expressed as minimizing the Frobenius norm of the matrix $A = X - uv^T$, over u, v. Since only the term uv^T counts, we can always impose $u^T u = 1$. With $a = [a_1, \ldots, a_m]$, where $a_i \in \mathbb{R}^n$, P_3 can be written as:

$$\begin{aligned} &\min_{A,v,u} \sum_{i=1}^{m} \|a_i\|_2^2 \ : \ x_i = v_i u + a_i, u^T u = 1 \\ = &\min_{v,u} \sum_{i=1}^{m} \|x_i - v_i u\|_2^2 \ : \ u^T u = 1 \end{aligned}$$

which is exactly P_2, as given by (1).

Exercise 2 (PCA and Senate Voting Data) We return to the Senate voting data examined in HW1, with X the $m \times n$ data matrix, where each row corresponds to a Senator, and each column to a bill.

1. Find a n-vector a and scalar b such that the variance of the corresponding score function $f(x) = a^T x + b$ is maximized.
2. How does the variance obtained previously compare to the one obtained with a set to the center of the data points, and b set so that the average score is zero? Comment on the phrase "senators vote according to the party average".
3. What is the total variance explained by the first two principal components? Plot the data projected on the corresponding plane.
4. Based on the first principal component, which bill(s) would you say have been the most important? Which Senators are the most "extreme"?

Solution 2

1. The linear function $f(x) = a^T x + b$ can be thought of as a projection of the data point x onto a hyperplane.
   We can think of $\mathbb{E}[f(x)]$ as $\frac{1}{n}\sum_{i=1}^{n} f(x_i)$.

   The variance of an RV is defined as

   $$\mathbb{E}[f(x)^2] - \mathbb{E}[f(x)]^2$$

   We can find

   $$\mathbb{E}[f(x)] = a^T \mathbb{E}[x] + b$$

   We can find the expecation squared

   $$\mathbb{E}f(x)^2 = \mathbb{E}[(a^T x_i + b)^2] = \mathbb{E}[(a^T x)^2] + 2b\mathbb{E}[a^T x] + b^2$$

   Hence the variance is

   $$\begin{aligned}&\mathbb{E}[(a^T x)^2] + 2b\mathbb{E}[a^T x] + b^2 - (a^T \mathbb{E}[x] + b)^2\\ &= a^T \mathbb{E}[xx^T]a + 2b\mathbb{E}[a^T x] + b^2 - (a^T \mathbb{E}[x]\mathbb{E}[x]^T a + 2b\mathbb{E}[a^T x] + b^2)\\ &= a^T(\mathbb{E}[xx^T] - \mathbb{E}[x]\mathbb{E}[x]^T)a\end{aligned}$$

   Note that

   $$\mathbb{E}[xx^T] = \frac{1}{n} X^T X$$

Let $\mathbb{E}[x] = \mu_x$
We get the variance to be

$$a^T \left(\frac{1}{n} X^T X - \mu_x \mu_x^T \right) a$$

This is familiar to the PCA objective, as we are trying to find:

$$\max_{a \,:\, a^T a = 1} \; a^T \left(\frac{1}{n} X^T X - \mu_x \mu_x^T \right) a$$

Note that we should require $a^T a = 1$ in order to avoid degenerate solutions. This problem can be solved by PCA, which we know how to do. Hence, a should be the first principal component, a_1 from the data and b can be set to any number.

On a side note, it is good to understand how maximizing the variance of the projection is equivalent to minimizing the reconstruction error in a lower dimensional space (both of which PCA does).

See Jupyter notebook for solution code.

2. Let a = μ. We want

$$\mathbb{E}[f(x)] = 0$$

Hence

$$\mathbb{E}[a^T x] = -b$$

$$-\mu^T \mu = b$$

The variance in this case is

$$\mu^T (\mathbb{E}[x x^T] - \mu \mu^T) \mu$$

See Jupyter notebook for solution code.

The variance for this vector is around 140 while the vector that explains best the data (first PC) has variance 150. Most of the variance is explained by the mean vector, which explains why senators vote according to the party average.

3. Now consider the first principal component, a_1 of

$$\frac{1}{n} X^T X - \mu_x \mu_x^T$$

This equivalent to the eigenvector of $\frac{1}{n}X^TX - \mu_x\mu_x^T$ with the largest eigenvalue, λ_1. We know that the variance is:

$$\begin{aligned}&a_1^T\left(\frac{1}{n}X^TX - \mu_x\mu_x^T\right)a_1\\ =&a_1^T\lambda_1 a_1\\ =&\lambda_1\end{aligned}$$

as a_1 is a unit vector. Consider $a = [a_1, a_2]$. Then the explained variance by both vectors becomes:

$$a_1^T\left(\frac{1}{n}X^TX - \mu_x\mu_x^T\right)a_1 + a_2^T\left(\frac{1}{n}X^TX - \mu_x\mu_x^T\right)a_2 = \lambda_1 + \lambda_2$$

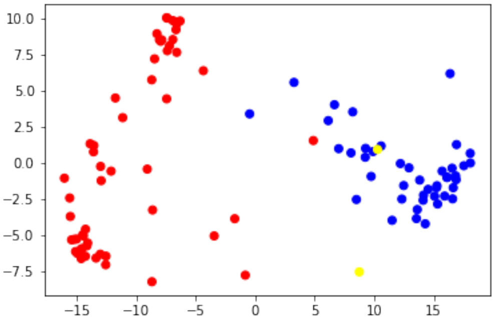


Figure 1: PC1 vs PC2

4. To measure the "importance" of bills, we can consider the first principal component an axis for how party-aligned a bill is. The more a bill is important, the closer it will be to the center of the principal component. Bills on the extremes will be more partisan, while central bills will be more non-partisan.

   We can look at the absolute value of the scores of the senators $f(x)$ to determine how extreme a senator is in their voting.

   See Jupyter notebook for solution code.

Exercise 3 (Generalized Eigenvalues and Image Segmentation)

In this exercise, we will implement an algorithm for image segmentation from a graph-theoretic approach. As a simplified problem, we will focus on separating the foreground of an image from its background. We provide you with a skeleton code in the archive image_segmentation.zip that imports an image and guide you through the implementation.

An image is described as a matrix M of shape (N, M) whose values represent the gray scale color normalized between zero and one. Even though we restrict now to gray images, please note that the same algorithm could be easily extended to colored images.

An undirected graph is a triplet (V, E, W) where V is the set of nodes, E the set of edges, and W the set of weights for each edge. As a first step, we would like to construct a graph that represents the image. For that purpose, we choose V to be the set of pixels. We say that there is an edge (i, j) between pixel i and pixel j, $1 \leq i, j \leq NM$, if they are neighbors in the image (adjacent in the matrix M). For each edge (i, j), we will use as weight $w_{ij} = 1 - abs(M_{x_i,y_i} - M_{x_j,y_j})$, where (x_i, y_i) represents the location of pixel i and (x_j, y_j) represents the location of pixel j. This is a measure of pixel similarity. Finally, we are interested in the whole affinity matrix W of shape (NM, NM), where each entry has the value zero is there is no edge between the pixels and the value w_{ij} if there is an edge.

1. Complete the code in the notebook `image-seg.ipynb` that transforms the image `img-cup-small.png` into the affinity matrix W.

Let D be the diagonal matrix of shape (NM, NM), where the i^{th} entry equals the sum of weights associated with edges incident on pixel i. That is, $D(i) = \sum_j W(i, j)$. We wish to solve the generalized eigenvalue problem $(D - W)y = \lambda Dy$ for all eigenvectors and all eigenvalues.

This eigenvalue problem is a known approximation to finding a partition of a graph that minimize the normalized cut in graph theory. Also, if the graph is connected, then the largest eigenvalue of the adjacency matrix as well as the smallest eigenvalue of the Laplacian have multiplicity 1. We can expect that the gap between this and the nearest eigenvalue is related to some kind of connectivity measure of the graph (for more information on the general topic of graph eigenvalues, one can refer to `http://web.cs.elte.hu/~lovasz/eigenvals-x.pdf`). The partition that approximate the solution to the normalized cut problem is derived from the generalized eigenvector with the second smallest eigenvalue and can be straightforwardly used in image segmentation.

2. Explain how to relate this problem to the symmetric eigenvalue problem we saw in class. Furthermore, explain how you would solve this problem provided routine code to compute a SVD.

3. Find the generalized eigenvector y for the eigenvector with the second smallest eigenvalue in the case of your given image. You will rely on linear algebra routine libraries like `numpy.linalg`.

The i-th entry of y can be viewed as a soft indicator of the component membership of the i-th pixel (foreground or background).

4. Use this indication to transform your image into a binary image (the so-called result of image segmentation). Try separating pixels corresponding to positive / negative values and above the median / below the median values of the entries of y to split the image into two components. Plot the obtained binary image and comment on your results.

Solution 3

1. See Jupyter notebook for solution code.
2. We have the generalized eigenvalue problem

$$(D - W)y = \lambda Dy.$$

Transform the system to standard eigenvalue problem shown in class

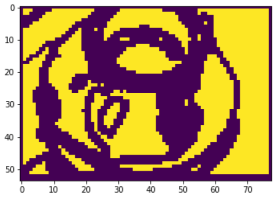
$$D^{-\frac{1}{2}}(D - W)D^{-\frac{1}{2}}z = \lambda z$$

$$z = D^{\frac{1}{2}}y$$

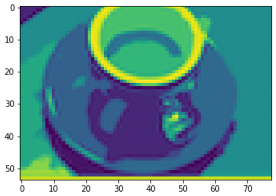
Matrix $(D - W)$ is the Laplacian matrix, and is positive semidefinite. Since D is a diagonal matrix of positive entries, it is positive definite. As a result, matrix $D^{-\frac{1}{2}}(D - W)D^{-\frac{1}{2}}$ is positive semidefinite and the system can be solved by taking the SVD of the matrix. We then can recover $y = D^{-\frac{1}{2}}z$.

3. We first compute the eigenvalue decomposition of the matrix $D^{-1/2}(D-W)D^{-1/2}$, and then we can recover the 2nd smallest eigenvector. See Jupyter notebook for solution code.

4.



Segmented Labels



Original image

See Jupyter notebook for our solution code.

Exercise 4 (Diet Planning)

We consider a set of n basic foods (such as rice, beans, apples) and a set of m nutrients or components (such as protein, fat, sugar, vitamin C). Food j has a cost given by c_j (say, in dollars per gram), and contains an amount N_{ij} of nutrient i (per gram). (The nutrients are given in some appropriate units, which can depend on the particular nutrient.) A daily diet is represented by an n-vector d, with d_i the daily intake (in grams) of food i.

1. Express the condition that a diet d contains exactly the total nutrient amounts given by the m-vector n_{des}, and has a total cost B (the budget) as a set of linear equations in the variables $d_1, ..., d_n$. (The entries of d must be nonnegative, but we ignore this issue here.)

2. Now suppose we are given $n+1$ new constraints that requires how different the daily diet d is from some predetermined set diet $s_i \in \mathbb{R}^n$, for $i = 1, ..., n+1$, using the 2-norm as a measure of difference. That is, we have

$$\|d - s_i\|_2 = p_i$$

   Express these constraints as a set of n constraints **linear** in d.

Solution 4

1. Note that N is a matrix of dimension m by n, and that column j of N corresponds to the nutrients in food j. We have:

$$\begin{aligned} Nd &= n_{des} \\ c^T d &= B. \end{aligned}$$

2. We square the 2-norm:

$$\begin{aligned} \|d - s_i\|_2 &= p_i \\ (d - s_i)^T (d - s_i) &= p_i^2 \\ d^T d - 2d^T s_i + s_i^T s_i &= p_i^2 \end{aligned}$$

   Then we remove the quadratic term by subtracting the $(n+1)$-th constraint from every other constraints. We have, as desired,

$$-2d^T s_i + s_i^T s_i + 2d^T s_{n+1} - s_{n+1}^T s_{n+1} = p_i^2 - p_{n+1}^2$$

Exercise 5 (A result related to Gaussian distributions)

Let $\Sigma \in S^n_{++}$ be a symmetric, positive definite matrix. Show that

$$\int_{\mathbb{R}^n} e^{-\frac{1}{2}x^T\Sigma^{-1}x} dx = (2\pi)^{\frac{n}{2}}\sqrt{\det\Sigma}.$$

You may assume known that the result holds true when $n = 1$. The above shows that the function $p = \mathbb{R}^n \to \mathbb{R}$ with (non-negative) values

$$p(x) = \frac{1}{(2\pi)^{n/2} \cdot \sqrt{\det\Sigma}} e^{-\frac{1}{2}x^T\Sigma^{-1}x}$$

integrates to one over the whole space. In fact, it is the density function of a probability distribution called the multivariate Gaussian (or normal) distribution, with zero mean and covariance matrix Σ.

$$\int_{x\in\mathbb{R}^n} f(x) = |\det P| \cdot \int_{z\in\mathbb{R}^n} f(Pz)dz.$$

Solution 5

When $n = 1$, the equation reads

$$\int_x e^{-\frac{1}{2}d^{-1}x^2} = \sqrt{2\pi d} \tag{2}$$

For the higher dimension case, we use the eigenvalue decomposition of Σ. Since $\Sigma = UDU^T$, we have

$$\Sigma^{-1} = UD^{-1}U^T \tag{3}$$

Substituting this in our original expression, we have

$$\begin{aligned}
\int_{\mathbb{R}^n} e^{-\frac{1}{2}x^TUD^{-1}U^Tx}dx &= \int_{\mathbb{R}^n} e^{-\frac{1}{2}x^TD^{-1}x}dx \\
&= \int_{\mathbb{R}^n} e^{-\frac{1}{2}(d_1x_1^2+\cdots+d_nx_n^2)}dx \\
&= (2\pi)^{\frac{n}{2}}\sqrt{d_1\cdots d_n} \\
&= (2\pi)^{\frac{n}{2}}\sqrt{\det\Sigma}
\end{aligned}$$

where $D = \text{diag}(d_1, ..., d_n)$. The first equality is due to the fact that left multiplication by U^T is a bijection from $\mathbb{R}^n$ to itself. We used the $n = 1$ case in the third equality, and the fact $\det(UDU^T) = \det(D)$ in the last equality.

Exercise 6 (Analysis of calendar days) Consider a matrix made up of numerical representations of a range $\mathcal{T}$ of calendar days. For example, the matrix below represents the range $\mathcal{T}$ starting on November 1, 2013 and ending on November 3, 2013:

$$X = \begin{pmatrix} 2013 & 11 & 1 \\ 2013 & 11 & 2 \\ 2013 & 11 & 3 \end{pmatrix}$$

1. Form a matrix X for the date range $\mathcal{T}$ starting on November 1, 2003 and ending on October 31, 2013.
2. Find an SVD of the matrix, and print out the right singular vectors and corresponding singular values.
3. What is the exact rank of X? Numerically, would you say that the rank of the matrix is "low"? Comment.
4. Plot the left singular vectors. Do these vectors exhibit a pattern? How would you interpret that pattern?
5. Explain the significance of the right singular vector v corresponding to the smallest singular value. How would you approximate such a vector by hand?
6. How would you expect the condition number (ratio from largest to smallest singular value) to behave as the number of calendar days in $\mathcal{T}$ increases?

Solution 6

For full details and source code, please consult `calendar_analysis_.ipynb`.

1.
$$X = \begin{bmatrix} 2003 & 11 & 1 \\ 2003 & 11 & 2 \\ & \vdots & \\ 2013 & 10 & 30 \\ 2013 & 10 & 31 \end{bmatrix}$$

2.

$$U = \begin{bmatrix} 0.01650054 & 0.02757517 & 0.02190335 \\ 0.01650061 & 0.02569526 & 0.02188012 \\ & \vdots & \\ 0.01658483 & -0.02866575 & 0.01625599 \end{bmatrix}$$

$$\Sigma = \mathbf{diag}(121388.22, 531.92, 208.48)$$

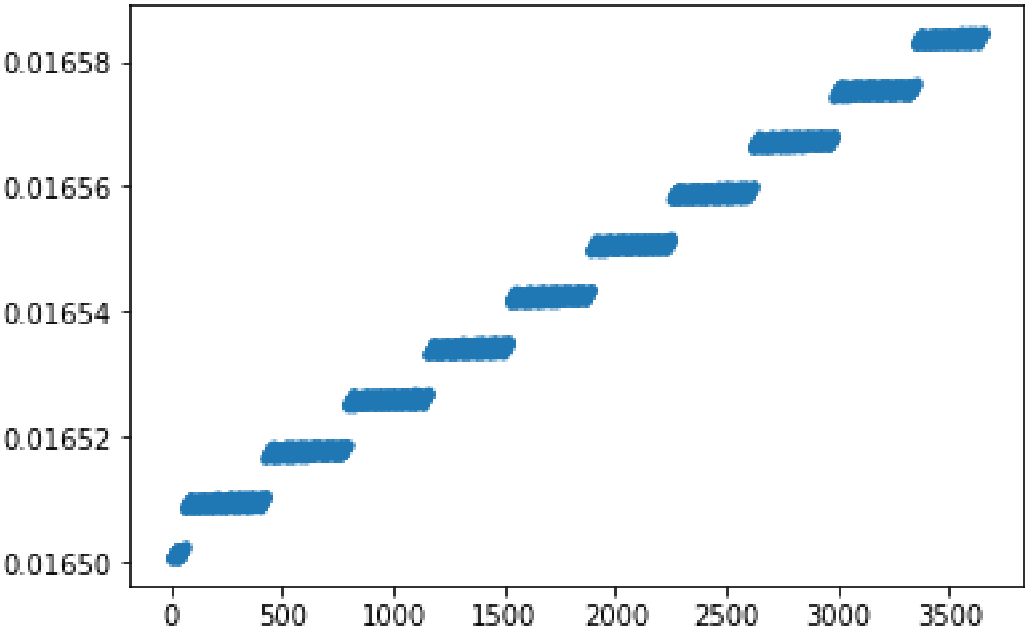
$$V^T = \begin{bmatrix} 0.99996405 & 0.00324731 & 0.00783294 \\ 0.00784854 & -0.00481823 & -0.99995759 \\ -0.00320943 & 0.99998312 & -0.00484354 \end{bmatrix}$$

The first eigenvalue corresponds to the first row of V^T, and so on.

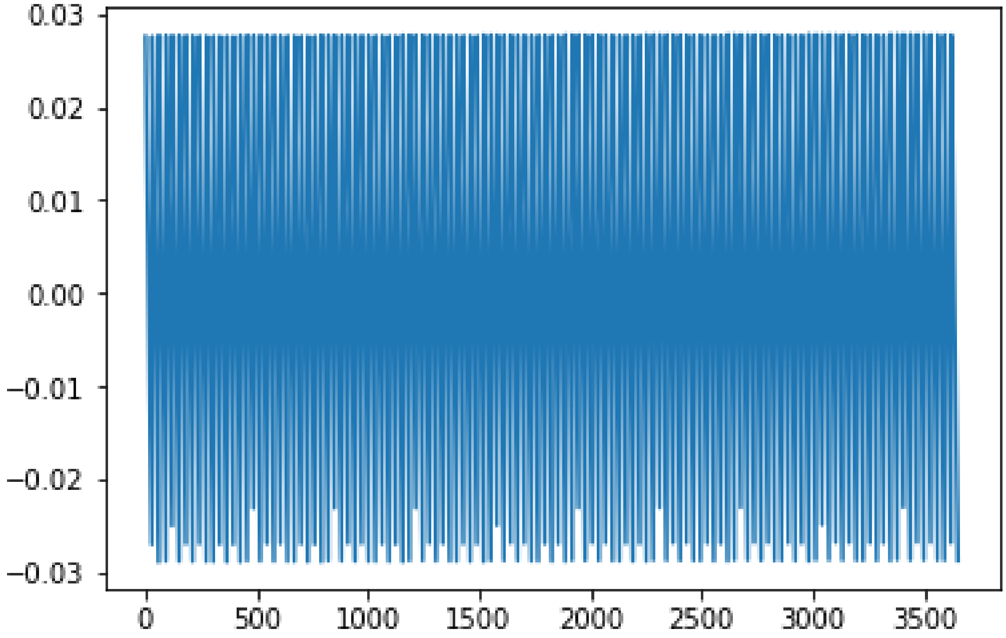
3. The rank is 3. Compared to the range of dates, this rank is low.

4. Plotting the left singular vectors, we note the following:

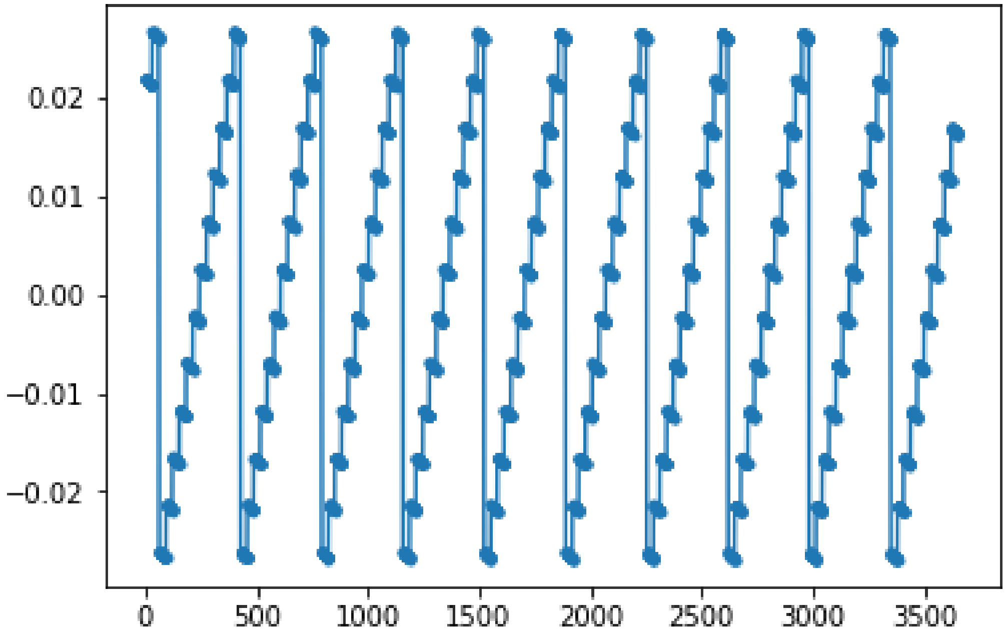
   (a) v_1 represents years, indicated by each step



   (b) v_2 represents days, with faster repetition



(c) v_3 represents months, indicated by 12 dots; the peaks correspond to years



5. From part (4), we know that the 3rd singular vector corresponds to months, which appears in the 2nd column of our data. This implies the right singular vector will be of the form $[0, 1, 0]^T$. The middle 1 places a emphasis on the month.

6. We expect the ratio to increase, since the maximal year will increase while the maximal month remain the same.