Section 08: Solutions

1. Singular Value Decomposition

(a) Let A be a symmetric $n \times n$ matrix. Take U to be a matrix whose columns are an orthonormal eigenbasis of A, and S to be a diagonal matrix of eigenvalues (ordered the same as U). Show that for any vector x, Ax is the same as USU^Tx (i.e. USU^T is the SVD of A)

Solution:

S Let $\{u_1, u_2, \ldots, u_n\}$ be the rows of U. They are orthogonal to each other and unit norm. For any $x \in \mathbb{R}^n$ we can write $x = \sum_{i=1}^n \alpha_i u_i$. Then we have:

$$\begin{aligned}
USU^Tx &= USU^T \sum_{i=1}^n \alpha_i u_i \\
&= \sum_{i=1}^n \alpha_i USU^T\ u_i \\
&= \sum_{i=1}^n \alpha_i USe_i \\
&= \sum_{i=1}^n \alpha_i U\lambda_i e_i \\
&= \sum_{i=1}^n \alpha_i \lambda_i u_i \\
&= \sum_{i=1}^n \alpha_i Au_i \\
&= A\sum_{i=1}^n \alpha_i u_i \\
&= Ax
\end{aligned}$$

(b) Let A have SVD USV^T. Show AA^T has the columns of U as eigenvectors with associated eigenvalues S^2.

Solution:

We have $A = USV^T$ then:

$$\begin{aligned}
AA^T &= USV^T(USV^T)^T \\
&= USV^T((V^T)^TS^TU^T) \\
&= USV^TVSU^T \\
&= USV^TVSU^T \\
&= USISU^T \\
&= US^2U^T
\end{aligned}$$

Since we can diagonalize AA^T into US^2U^T, it has eigenvectors that are columns of U and associated eigenvalues S^2.

(c) Let A have SVD USV^T. Show A^TA has the columns of V as eigenvectors with associated eigenvalues S^2.

Solution:

We have $A = USV^T$ then:

$$\begin{aligned} A^TA &= (USV^T)^TUSV^T \\ &= VSU^TUSV^T \\ &= VSISV^T \\ &= VS^2V^T \end{aligned}$$

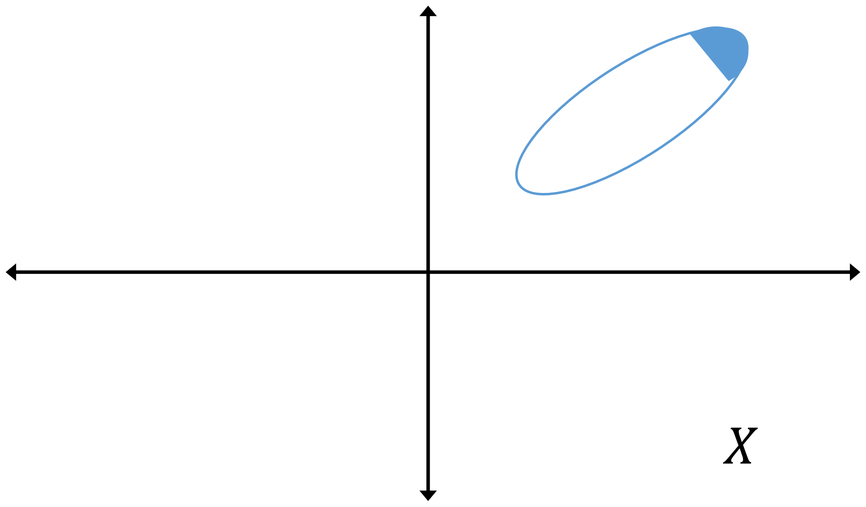
Since we can diagonalize A^TA into VS^2V^T, it has eigenvectors that are columns of V and associated eigenvalues S^2.

2. SVD Pictures

We've seen before that demeaning our data makes it easier to work with. There's a more general operation called "whitening" where we also normalize the important directions of our data. In this problem, we'll do the operations corresponding to one version of whitening as a way to get better intuition on how SVD works.

Let $X \in \mathbb{R}^{n\times d}$ be a matrix of data points, and J be $\mathbf{I} - \mathbf{1}\mathbf{1}^T/n$. Let JX have a singular value decomposition of $JX = USV^T$.

Suppose we know that our points were drawn from a Gaussian distribution with covariance Σ. We would expect most of our points to lie in an ellipse, whose axes are the eigenvectors of Σ, scaled by the corresponding eigenvectors. We've drawn that ellipse below, with one area shaded so we can see its orientation.



For each of the following matrices:

- Verify that the resulting matrix is still $n \times d$, and therefore can be interpreted as modifying the datapoints of X
- Draw what the resulting data set would look like (i.e. how would the ellipse representing the covariance move?).

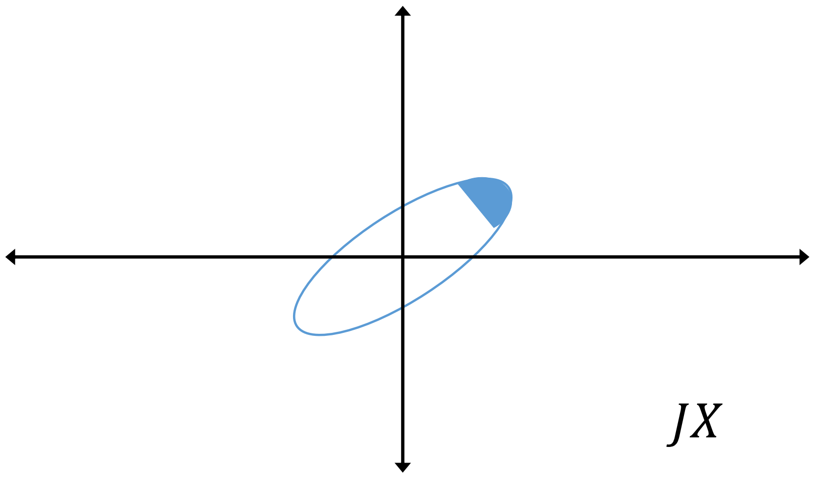
(a) JX **Solution:**

Since J is $n \times n$, the resulting matrix is, indeed $n \times d$.

Notice that $\mathbf{1}\mathbf{1}^T/n$ is an $n \times n$ matrix where every entry is $1/n$.

$$\begin{aligned} JX &= (\mathbf{I} - \mathbf{1}\mathbf{1}^T/n)X \\ &= X - \mathbf{1}\mathbf{1}^T X/n \\ &= X - \mathbf{1}\left(\sum_{i=1}^{n} x_i^T/n\right) \\ &= X - \mathbf{1}\overline{x} \end{aligned}$$

where $\overline{x}$ is the average of the rows of X. Thus JX is just X "demeaned"



(b) JXV **Solution:**

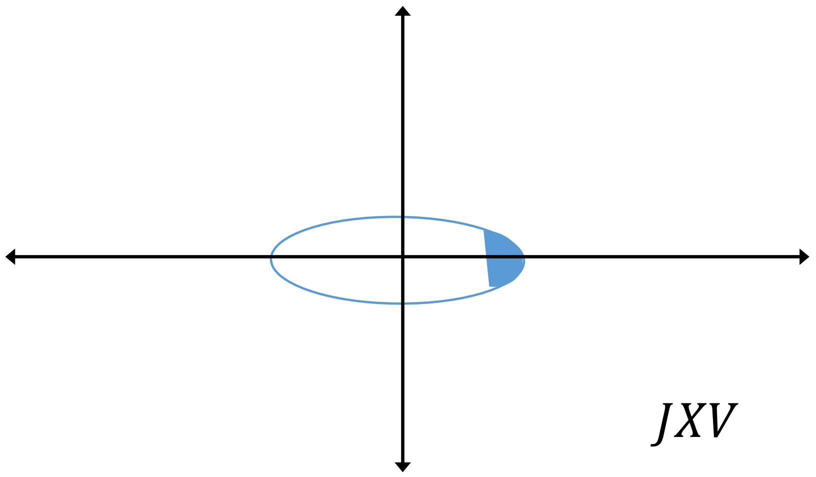
Since V is $d \times d$, JXV is still $n \times d$.

Below we will give a few different views of the calculation (they all say the same thing, but alternative wordings may give a different perspective)

We want to think about row i of JXV. Define y_i to be the i^{th} row of JX written as a column vector. When we multiply JXV, we will find $y_i^T v_j$ and put that number in entry i, j of the matrix. Since the columns of V form an orthnormal basis of $\mathbb{R}^d$, the i^{th} row of JXV is just y rewritten in the basis of V.

Said differently, if row i of JXV is the vector z then $y_i = \sum z[j]v_j$.

But then what does JXV look like? Well the j^{th} entry of row i is its dot product with v_j. I.e. in each direction of the standard basis we are going to go as far as we went in the principal component directions in y. So we have rotated the ellipse to now be on the standard basis.



(c) $JXVS^{-1}$, where S^{-1} is the $d \times d$ diagonal matrix, where $S^{-1}_{i,i} = 1/S_{i,i}$ (note that since S is not square, calling a matrix S^{-1} is an abuse of notation)

Solution:

Since S^{-1} is $d \times d$, the matrix remains $n \times d$.

Note that S^{-1} just renormalizes each column j by $1/\sigma_j$. Thus row i of $JXVS^{-1}$ is the i^{th} demeaned data point, written in the singular vector basis, now normalized, so the most extreme data points have length at most 1 in the new basis.

How do we know the lengths become at most 1? The easiest way is to look at a single row, let e_i be the vector with a 1 in entry i and all 0's everywhere else. Note that $e_i^T A$ is the i^{th} row of the matrix A. To understand the length of the i^{th} row we want:

$$\begin{aligned} ||(e_i^T JXVS^{-1})||_2^2 &= (e_i^T JXVS^{-1})^T (e_i^T JXVS^{-1}) \\ &= (e_i^T USV^T VS^{-1})^T (e_i^T USV^T VS^{-1}) \\ &= (e_i^T USS^{-1})^T (e_i^T USS^{-1}) \\ &= (e_i^T UI')^T (e_i^T UI') \end{aligned}$$

Where I' is the $n \times d$ matrix, which has 1s in every entry on the diagonal and 0's everywhere else. (Recall that S^{-1} isn't really an inverse – S isn't square so it can't have a real inverse)

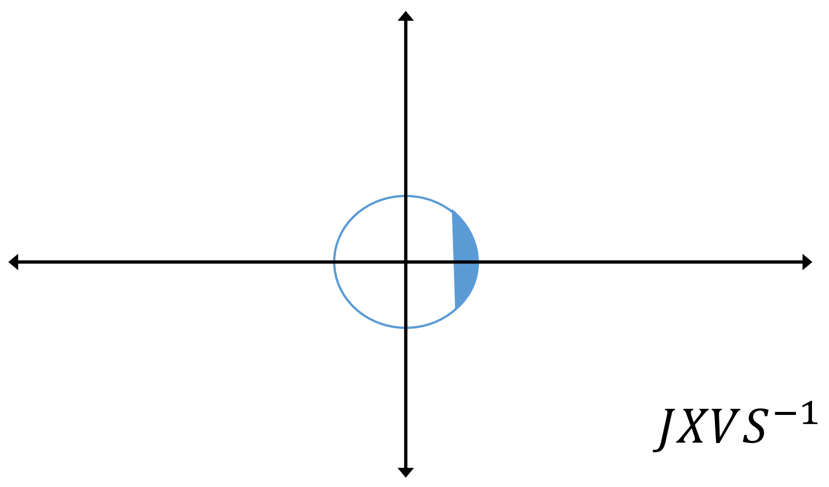
Let U' be the $n \times d$ matrix formed by deleting columns $d+1, d+2, \ldots, n$ of U.

$$\begin{aligned} (e_i^T UI')^T (e_i^T UI') &= (e_i^T U')^T (e_i^T U') \\ &= U_i'^T U_i' \leq 1 \end{aligned}$$

Where the last inequality follows from the fact that (the full row) U_i is an orthonormal vector. Since we've just deleted entries from it, the length of the vector only decreased, and so the length is still at most 1.

Vectors of length at most 1 lie inside a circle, so we've "squashed" our vectors. Notice that since we're shrinking each direction according to its singular value, we are shrinking each vector by a different amount, such that we end up with a circle.

What's the radius of our circle? It turns out it's about $\frac{1}{\sqrt{n}}$ – try drawing some real data for various n and see what happens!



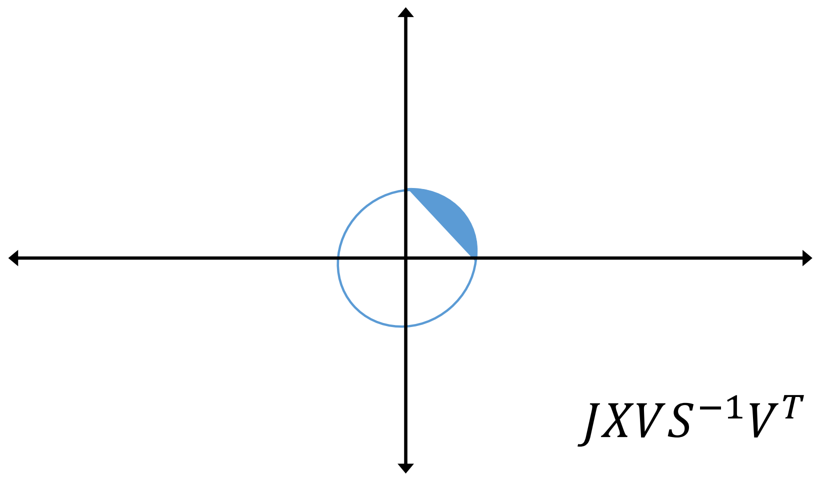
(d) $JXVS^{-1}V^T$

Solution:

row i of $JXVS^{-1}V^T$: is harder to understand than the previous ones, let's do a calculation. Recall that entry i, j of $JXVS^{-1}$ is $\frac{1}{\sigma_j} y_i^T v_j$, where y_i is the demeaned version of x_i.

Then entry i, j of $JXVS^{-1}V^T$ is: $\sum_{k=1}^{d} \frac{1}{\sigma_k} y_i^T v_k v_k[j]$

so we can write row i as: $\sum_{k=1}^{d} \frac{1}{\sigma_k} y_i^T v_k v_k^T = \sum_{k=1}^{d} \frac{1}{\sigma_k} y_i I$ So we have just y_i but normalized by the σs. Thus we have "rotated" the points back to the original space, but kept them the same length as before.



3. Extra Material: Kernel Principal Component Analysis

This material here is intended as a way to get more practice with both PCA and kernels – we won't expect you to have a deep understanding of kernel PCA as a process.

We've seen that kernels are a way of using complicated feature maps without actually calculating the feature maps, as long as:

- We only care about inner products in the feature space.
- The new feature space is chosen so that we can use the "kernel trick" to calculate dot products directly, without actually applying the feature map.

Since we know:

- PCA is useful in settings where we have linear relationships in our data.
- feature maps can let us use linear methods even when our data has non-linear relationships

it's a reasonable idea to think we might try to apply a feature map, and do PCA in the feature space. But we've seen that we can use very powerful feature maps if all we care about is inner products (via the kernel trick).

In this problem we'll see that this is possible. Let $\{x_i\}_{i=1}^n$ be a set of datapoints in d-dimensions, already demeaned. Define $\Sigma = \frac{1}{n}\sum_{i=1}^n x_i x_i^T$ to be the $(d \times d)$ emperical covariance matrix of the data points.

(a) Let $\phi : \mathbb{R}^d \to \mathbb{R}^k$ be a feature map. What is the formula for Ψ, the emperical covariance matrix of $\{\phi(x_i)\}_{i=1}^n$? What are its dimensions?

Solution:

$\Psi = \frac{1}{n}\sum_{i=1}^n \phi(x_i)\phi(x_i)^T$ The matrix is $k \times k$.

(b) Let v_j be an eigenvector of Ψ. Show that Ψv_j can be written as a linear combination of $\phi(x_i)$.

Solution:

$\Psi v_j = \frac{1}{n}\sum_{i=1}^n \phi(x_i)\phi(x_i)^T v_j$ Regrouping $\phi(x_i)^T v_j$ as a coefficient a_{ij}, we get:

$$\Psi v_j = \frac{1}{n}\sum_{i=1}^n a_{ij}\phi(x_i)$$

So we have written $\lambda_j v_j$ as a linear combination of the $\phi(x_i)$.

(c) Use the observation from the previous part to write the following equation:

$$\frac{1}{n}\sum_{i=1}^n \phi(x_i)\phi(x_i)^T \sum_{i=1}^n a_{ij}\phi(x_i) = \lambda_j \sum_{i=1}^n a_{ij}\phi(x_i)$$

where a_{ij} are scalars.

Solution:

Expand the definition of Ψ, and we get the equation above.

(d) Rewrite the equation above so that it uses the kernel matrix (Hint: left-multiply by $\phi(x_\ell)^T$ first).

Solution:

Let's start with $\phi(x_\ell)^T$ times the left hand side of the equation above:

$$\begin{aligned}\phi(x_\ell)^T \frac{1}{n}\sum_{i=1}^n \phi(x_i)\phi(x_i)^T \sum_{i=1}^n a_{ij}\phi(x_i) &= \frac{1}{n}\sum_{i=1}^n \phi(x_\ell)^T\phi(x_i)\phi(x_i)^T \sum_{k=1}^n a_{kj}\phi(x_k)\\ &= \frac{1}{n}\sum_{i=1}^n \phi(x_\ell)^T\phi(x_i) \sum_{k=1}^n a_{kj}\phi(x_i)^T\phi(x_k)\\ &= \frac{1}{n}\sum_{i=1}^n k(x_\ell, x_i) \sum_{k=1}^n a_{kj}k(x_i, x_k)\end{aligned}$$

On the right hand side:

$$\phi(x_\ell)^T \lambda_j \sum_{i=1}^n a_{ij}\phi(x_i) = \lambda_j \sum_{i=1}^n a_{ij}k(x_\ell, x_i)$$

(e) Use the previous equation to generate a matrix version of this equation (Hint: what would happen if you replaced $\phi(x_\ell)$ with each x_i in order and stacked the equations?)

Solution:

Notice that the ℓ^{th} row of K^2 is the ℓ^{th} row of K, dotted with each column of K. Thus the left hand side is exactly the ℓ^{th} row of $K^2 a_j$, where a_j is the vector formed by all the a_{ij}. Thus if we move the $\frac{1}{n}$ to the other side, and stack all possible values of ℓ, we get a LHS of $K^2 a_j$. On the right hand side, the summation is the ℓ^{th} row of K time a_j, again stacking all possible ℓ on top of each other (and remembering we moved the n to the other side we have:

$$K^2 a_j = \lambda_j n K a_j$$

Eliminating one K from each side (as we did for regular PCA) we have

$$K a_j = \lambda_j n a_j$$

(f) Argue that if we have a ϕ that allows for the "kernel trick" everything we need to do for PCA (i.e. finding X, a_j and calculating $\phi(x)^T v_j$ for any x and v_j can always be done without going into the kernel space (i.e. using only the kernel function).

Solution:

To solve the eigenvalue problem, we need to calculate K (which we can do since we assumed we can do the kernel trick), which allows us to calculate the a_j.

we also will want to calculate the $\phi(x)^T v_j = \sum_{i=1}^n a_{ij}\phi(x)^T\phi(x_i) = \sum_{i=1}^n a_{ij}k(x, x_i)$ so we can use the kernel trick here as well.