Mathematical Methods for Machine Learning and Signal Processing SS 2019

Lecture 18: Principal Component Analysis

V. Morgenshtern (compiled from notes by M. Deisenroth, S. Zafeiriou, and A. Ng)

Agenda:

1. Singular value decomposition (SVD)
2. Dimensionality reduction and SVD
3. Principal component analysis (PCA)
4. Link between SVD and PCA

1 Singular value decomposition

Recall that if $\mathbf{A} \in \mathbb{R}^{m\times n}$, then there exist orthogonal matrices

$$\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_m] \in \mathbb{R}^{m\times m} \text{ and } \mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_m] \in \mathbb{R}^{n\times n}$$

such that

$$\mathbf{U}^\mathsf{T}\mathbf{A}\mathbf{V} = \mathbf{\Sigma} = \operatorname{diag}(\sigma_1, \dots, \sigma_p) \in \mathbb{R}^{m\times n},\ p = \min(m, n)$$

and

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

where $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_p \geq 0$ and $\operatorname{diag}(\sigma_1, \dots, \sigma_p)$ is a not-necessarily square matrix with $\sigma_1, \dots, \sigma_p$ on the main diagonal and zeros everywhere else. σ_i are called singular values of $\mathbf{A}$ and the vectors $\mathbf{u}_i$ and $\mathbf{v}_i$ are the corresponding ith left and right singular vectors, respectively.

The decomposition in (1) is called the Singular Value Decomposition (SVD) of $\mathbf{A}$.

If $r = \operatorname{rank}\mathbf{A}$, then

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r \geq \sigma_{r+1} = \dots = \sigma_p = 0$$

and the SVD decomposition of $\mathbf{A}$ can be written as

$$\mathbf{A} = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^T. \tag{2}$$

2 Dimensionality reduction and SVD

If the data can be modeled by an approximately low-rank model, then SVD can be used for dimensionality reduction on the data. This application is based on the following result.

Theorem 1. Let the SVD of $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\mathsf{T}$. If $k < r = \text{rank}(\mathbf{A})$ and

$$\mathbf{A}_k = \sum_{i=1}^{k} \sigma_i \mathbf{u}_i \mathbf{v}_i^\mathsf{T},$$

then

$$\min_{\mathbf{B}:\text{rank}\,\mathbf{B}=k} \|\mathbf{A} - \mathbf{B}\|_{op} = \|\mathbf{A} - \mathbf{A}_k\|_{op} = \sigma_{k+1}$$

where $\|\mathbf{A}\|_{op}$ denotes the operator norm of $\mathbf{A}$ that is equal to its largest singular value.

Proof. Since $\mathbf{U}^\mathsf{T}\mathbf{A}_k\mathbf{V} = \text{diag}\,(\sigma_1, \ldots, \sigma_k, 0, \ldots, 0)$, in follows $\text{rank}(\mathbf{A}_k) = k$ and $\mathbf{U}^\mathsf{T}(\mathbf{A} - \mathbf{A}_k)\mathbf{V} = \text{diag}\,(0, \ldots, 0, \sigma_{k+1}, \ldots \sigma_p)$. Therefore, $\|\mathbf{A} - \mathbf{A}_k\| = \sigma_{k+1}$.

Now suppose $\mathbf{B} \in \mathbb{R}^{m\times n}$ and $\text{rank}\,\mathbf{B} = k$. It follows that there is an orthonormal basis $\{\mathbf{q}_1, \ldots, \mathbf{q}_{n-k}\}$ so that $\mathcal{N}(\mathbf{B}) = \text{span}\{\mathbf{q}_1, \ldots, \mathbf{q}_{n-k}\}$. It follows that

$$\text{span}\{\mathbf{q}_1, \ldots, \mathbf{q}_{n-k}\} \cap \text{span}\{\mathbf{v}_1, \ldots, \mathbf{v}_{k+1}\} \neq \{\mathbf{0}\}.$$

This means that there exist unit $l2$-norm vectors $\mathbf{z}$ so that $\mathbf{B}\mathbf{z} = \mathbf{0}$ which can be written as a linear combination $\mathbf{z} = \sum_{i=1}^{k+1} \alpha_i \mathbf{v}_i$ with $\sum_{i=1}^{k+1} \alpha_i^2 = 1$ and $\alpha_i = \mathbf{v}_i^\mathsf{T}\mathbf{z}$. Since $\mathbf{B}\mathbf{z} = \mathbf{0}$ and, as follows from (2)

$$\mathbf{A}\mathbf{z} = \sum_{i=1}^{k+1} \sigma_i (\mathbf{v}_i^\mathsf{T}\mathbf{z})\mathbf{u}_i$$

we have

$$\|\mathbf{A} - \mathbf{B}\|_{op}^2 \geq \|(\mathbf{A} - \mathbf{B})\mathbf{z}\|_2^2 = \|\mathbf{A}\mathbf{z}\|_2^2 = \sum_{i=1}^{k+1} \sigma_i^2 (\mathbf{v}_i^\mathsf{T}\mathbf{z})^2 \geq \sigma_{k+1}^2 \sum_{i=1}^{k+1} \alpha_i^2 = \sigma_{k+1}^2$$

which completes the proof. □

We can use the theorem to reduce dimensionality of data. Assume we have n data points $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}$, $\mathbf{x}^{(i)} \in \mathbb{R}^m$. Let's stack the samples as columns of $\mathbf{X} = [\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}]$. Then, the best rank-$k$ approximation of the data, $\mathbf{X}$, is given by

$$\mathbf{X}_k = \mathbf{U}_k \Sigma_k \mathbf{V}_k^\mathsf{T}.$$

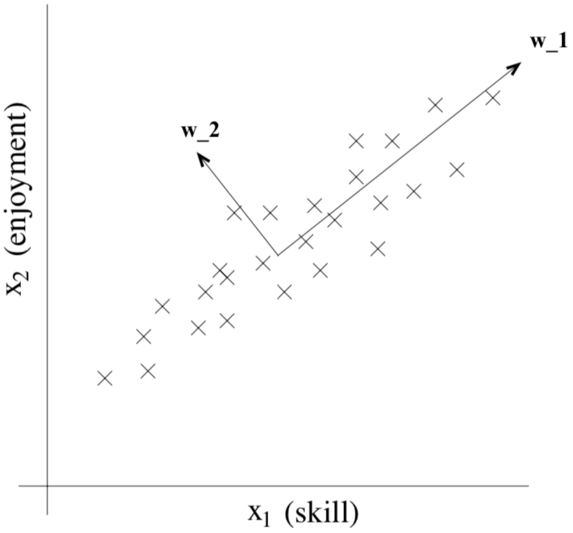
The approximation is composed of the k largest singular vectors of $\mathbf{X}$ and the corresponding singular vectors. When does this approximation makes sense?

3 Principal component analysis (PCA)

The approximation makes sense, if there are reasons to believe that the smallest singular values and the corresponding singular vectors may correspond to noise.

For example, suppose that $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}$, $\mathbf{x}^{(i)} \in \mathbb{R}^m$ correspond to attributes of n different types of automobiles, such as their maximum speed, turn radius, and so on. Let $\mathbf{x}^{(i)} \in \mathbb{R}^m$ for each i with $m \ll n$. But unknown to us, two different attributes, $x_k^{(i)}$ and $x_l^{(i)}$, give a car's maximum speed measured in miles per hour, and the maximum speed measured in kilometers per hour. These two attributes are therefore almost linearly dependent, up to only small differences introduced by rounding off to the nearest mph or kph. Thus, the data really lies approximately on an $n-1$ dimensional subspace. Can we automatically detect and remove this redundancy?

A more realistic example may be the following. Consider a dataset resulting from a survey of pilots for radio-controlled helicopters, where $x_1^{(i)}$ is a measure of the piloting skill of pilot i, and $x_2^{(i)}$ captures how much he/she enjoys flying. Because RC helicopters are very difficult to fly, only the most committed students, ones that truly enjoy flying, become good pilots. So, the two attributes, x_1 and x_2 are strongly correlated. Indeed, we might posit that the data actually lies along some diagonal axis (the $\mathbf{w}_1$ direction) capturing the intrinsic piloting "karma" of a person, with only a small amount of noise lying off the axis:



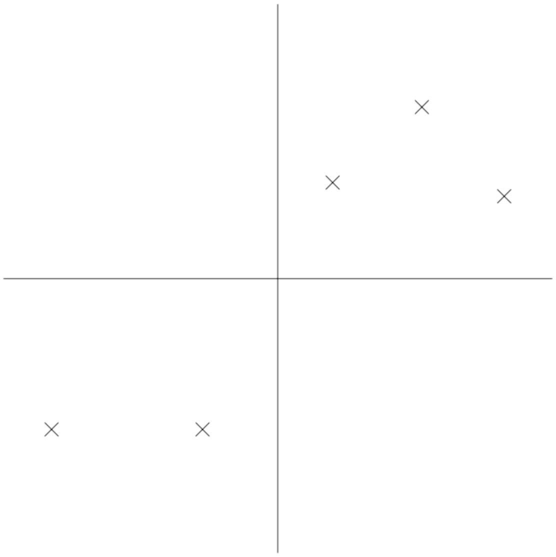
How can we automatically compute this $\mathbf{w}_1$ direction? This can be done using the PCA algorithm. Prior to running PCA, we typically standardize the data as follows.

1. Compute the mean data vector $\mu = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}^{(i)}$.
2. Replace each $\mathbf{x}^{(i)}$ with $\mathbf{x}^{(i)} - \mu$.
3. Compute the sample variance of each feature in the data $\sigma_j^2 = \frac{1}{n}\sum_{i=1}^{n}(x_j^{(i)})^2$.
4. Replace each $x_j^{(i)}$ with $x_j^{(i)}/\sigma_j$.

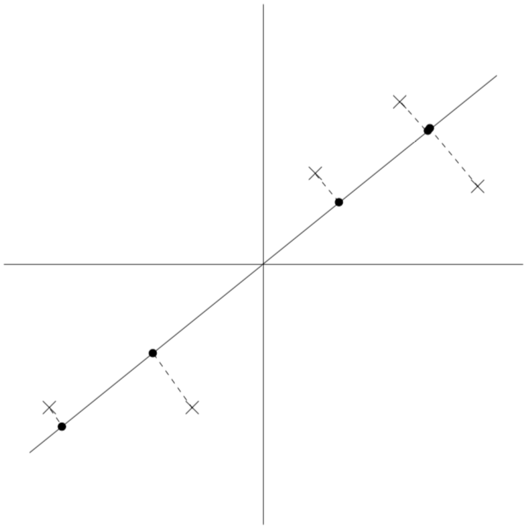
Steps 1 and 2 rescale the data to have zero mean. Steps 3 and 4 rescale each feature to have unit variance, which ensures that all features are treated on the same scale. For instance, if $x_1^{(i)}$ is the maximum speed of a car in kmh and $x_2^{(i)}$ is the number of seats in the car (2 to 4), then steps 3 and 4 rescale these two attributes to make them comparable.

After normalization, how do we compute the *major axis of variation* $\mathbf{w}$ of the data, that is the direction on which the data approximately lies? One way to pose this problem is as finding the unit vector **w** so that when the data is projected onto the direction corresponding to **w**, the variance of the projected data is maximized. Intuitively, the data starts off with some amount of variance/information in it. We would like to choose a direction, **w** so that if we were to approximate the data as lying in the direction/subspace corresponding to **w**, as much as possible of this variance is still retained.

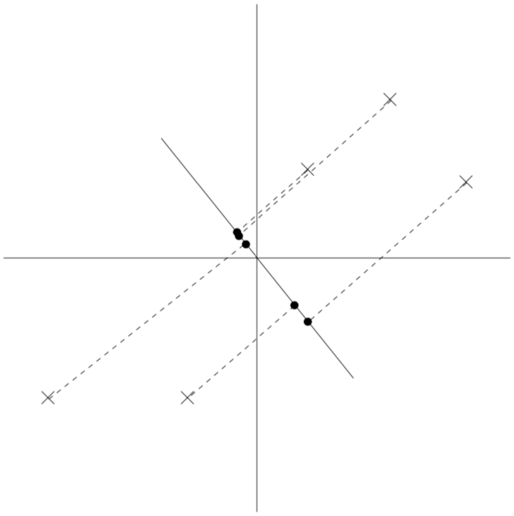
Consider the following dataset, on which we have already carried out the normalization steps:



Now, suppose we pick **w** as follows:



The circles denote the projection of the original data onto this line. We see that the projected data still has a fairly large variance, and the points tend to be far from the origin. In contrast suppose we had instead picked the following direction:



Here, the projections have a significantly smaller variance, and are much closer to the origin.

We would like to automatically select the direction $\mathbf{w}$ corresponding to the first of the two figures shown above. To formalize this, note that given a unit vector $\mathbf{w}$ and a point $\mathbf{x}$, the length of the projection of $\mathbf{x}$ onto $\mathbf{w}$ is given by $\mathbf{x}^\mathsf{T}\mathbf{w}$. Therefore, if $\mathbf{x}^{(i)}$ is a point in our dataset (one of the crosses in the plot), then its projection onto $\mathbf{w}$ (the corresponding circle in the figure) is distance $(\mathbf{x}^{(i)})^\mathsf{T}\mathbf{w}$ from the origin. Hence, to maximize the variance of the projections, we would like to choose a unit-length $\mathbf{w}$, $\|\mathbf{w}\|_2 = 1$, so as to maximize:

$$\begin{aligned}
\frac{1}{n}\sum_{i=1}^{n}((\mathbf{x}^{(i)})^\mathsf{T}\mathbf{w})^2 &= \frac{1}{n}\sum_{i=1}^{n}\mathbf{w}^\mathsf{T}\mathbf{x}^{(i)}(\mathbf{x}^{(i)})^\mathsf{T}\mathbf{w} \\
&= \mathbf{w}^\mathsf{T}\left(\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}^{(i)}(\mathbf{x}^{(i)})^\mathsf{T}\right)\mathbf{w} \\
&= \mathbf{w}^\mathsf{T}\underbrace{\frac{1}{n}\mathbf{X}\mathbf{X}^\mathsf{T}}_{\mathbf{S}}\mathbf{w}
\end{aligned}$$

where $\mathbf{S}$ is called the covariance matrix of the data.

We need to solve the following optimization problem:

$$\mathbf{w} = \arg\max_{\hat{\mathbf{w}}} \hat{\mathbf{w}}^\mathsf{T}\mathbf{S}\hat{\mathbf{w}} \text{ subject to } \hat{\mathbf{w}}^\mathsf{T}\hat{\mathbf{w}} = 1. \tag{3}$$

This is a constraint optimization problem. To solve it, we can compute the Lagrangian

$$L(\hat{\mathbf{w}}, \lambda) = \hat{\mathbf{w}}^\mathsf{T}\mathbf{S}\hat{\mathbf{w}} - \lambda(\hat{\mathbf{w}}^\mathsf{T}\hat{\mathbf{w}} - 1)$$

and set its gradient to zero:

$$\nabla_{\hat{\mathbf{w}}} L(\hat{\mathbf{w}}) = \mathbf{0} \Leftrightarrow \mathbf{S}\hat{\mathbf{w}} - \lambda\hat{\mathbf{w}} = \mathbf{0}$$

to obtain

$$\mathbf{S}\hat{\mathbf{w}} = \lambda\hat{\mathbf{w}}. \tag{4}$$

This means that $\mathbf{w}$ is an eigenvector of $\mathbf{S}$ and the Lagrangian multiplier λ is the corresponding eigenvalue. $\mathbf{S}$ is a symmetric positive semi-definite matrix, hence all of its eigenvalues are non-negative. What pair of eigenvalue and eigenvector should we choose? Plugging (4) into (5) we find that $\hat{\mathbf{w}}^\mathsf{T}\mathbf{S}\hat{\mathbf{w}} = \lambda^2$ is maximized if we choose λ to be the largest eigenvalue of $\mathbf{S}$, and $\hat{\mathbf{w}}$ is the corresponding eigenvector. This vector is also called the principal axis of the data.

More often than not we want to find more than one feature. This is done recursively. Suppose that we have already found the directions $\mathbf{w}_1, \ldots, \mathbf{w}_{d-1}$. Now we want to find new directions onto which we project the original data, $\mathbf{w}_d$, such that it is orthogonal to all previous directions and the variance of the projection onto $\mathbf{w}_d$ is as large as possible. The optimization problem we want to solve now becomes:

$$\mathbf{w}_d = \arg\max_{\hat{\mathbf{w}}_d} \hat{\mathbf{w}}_d^\mathsf{T}\mathbf{S}\hat{\mathbf{w}}_d \text{ subject to } \hat{\mathbf{w}}_d^\mathsf{T}\hat{\mathbf{w}}_d = 1,\ \mathbf{w}_i^\mathsf{T}\hat{\mathbf{w}}_d = 0 \text{ for } i < d. \tag{5}$$

The Lagrangian of this problem is:

$$L(\hat{\mathbf{w}}, \lambda) = \hat{\mathbf{w}}_d^\mathsf{T}\mathbf{S}\hat{\mathbf{w}}_d - \lambda_d(\hat{\mathbf{w}}_d^\mathsf{T}\hat{\mathbf{w}}_d - 1) - \sum_{i=1}^{d-1}\lambda_i\hat{\mathbf{w}}_d^\mathsf{T}\mathbf{w}_i$$

and set its gradient to zero:

$$\nabla_{\hat{\mathbf{w}}_d} L(\hat{\mathbf{w}}_d) = \mathbf{0}$$

and therefore

$$\mathbf{S}\hat{\mathbf{w}}_d - \lambda_d\hat{\mathbf{w}}_d - \sum_{i=1}^{d-1}\lambda_i\mathbf{w}_i = \mathbf{0}. \tag{6}$$

We will prove by induction that $\mathbf{w}_i$ are eigenvectors of $\mathbf{S}$ for all $1 \leq i \leq d$. Assume that $\mathbf{w}_i$ are eigenvectors of $\mathbf{S}$ for $1 \leq i < d$. Let's prove that $\mathbf{w}_d$ also is an eigenvector of $\mathbf{S}$. From the assumption that $\mathbf{w}_i$ are eigenvectors of $\mathbf{S}$ for all $1 \leq i \leq d$ we can derive that $\lambda_i = 0$ for $1 \leq i < d$ as follows. Multiply (7) by $\mathbf{w}_j^\mathsf{T}$ on the left:

$$\underbrace{\mathbf{w}_j^\mathsf{T}\mathbf{S}\hat{\mathbf{w}}_d}_{\lambda(\mathbf{S})_j\mathbf{w}_j^\mathsf{T}\hat{\mathbf{w}}_d=0} - \lambda_d\underbrace{\mathbf{w}_j^\mathsf{T}\hat{\mathbf{w}}_d}_{0} - \sum_{i=1}^{d-1}\lambda_i\underbrace{\mathbf{w}_j^\mathsf{T}\mathbf{w}_i}_{\delta_{ij}} = \mathbf{0},$$

so that $\lambda_i = 0$ for $1 \leq i < d$. Therefore, (7) is equivalent to

$$\mathbf{S}\hat{\mathbf{w}}_d = \lambda_d\hat{\mathbf{w}}_d, \tag{7}$$

we conclude that $\mathbf{w}_d$ is an eigenvector of $\mathbf{S}$, which completes the induction step. Applying exactly the same argument as before, by induction, we conclude that the eigenvalue corresponding to $\mathbf{w}_d$ is the d-th largest eigenvalue of $\mathbf{S}$.

To conclude, PCA transform looks for d orthogonal direction vectors (known as the principle axes) such that the projection of input sample vectors onto the principle directions has the maximal spread, or equivalently that the variance of the output coordinates is maximal. The principal directions are the first (with respect to descending eigenvalues) d eigenvectors of the covariance matrix $\mathbf{X}\mathbf{X}^\mathsf{T}$.

4 PCA as optimal linear approximation

Assume that we are trying to find the optimal way to approximate the data matrix $\mathbf{X}$ using only a small set of d vectors $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_d]$. The vectors in $\mathbf{W}$ are normalized as $\|\mathbf{w}_i\|_2 = 1$ but otherwise arbitrary.

We are trying to approximate each $\mathbf{x}_i$ using the columns of $\mathbf{W}$, i.e. $\mathbf{x}_i \approx \mathbf{W}\mathbf{y}_i = \tilde{\mathbf{x}}_i$. The optimal coefficients are given by least squares: $\mathbf{y}_i = (\mathbf{W}^\mathsf{T}\mathbf{W})^{-1}\mathbf{W}^\mathsf{T}\mathbf{x}_i$. Therefore, $\tilde{\mathbf{x}}_i = \mathbf{W}(\mathbf{W}^\mathsf{T}\mathbf{W})^{-1}\mathbf{W}^\mathsf{T}\mathbf{x}_i = \mathbf{W}\mathbf{W}^\mathsf{T}\mathbf{x}_i$, where we used that $\|\mathbf{w}_i\|_2 = 1$.

The optimal $\mathbf{W}$ is the solution of

$$\begin{aligned}
\mathbf{W} &= \arg\min_{\hat{\mathbf{W}}} \frac{1}{2}\sum_{i=1}^{n}\|\mathbf{x}_i - \tilde{\mathbf{x}}_i\|_2^2 \\
&= \arg\min_{\hat{\mathbf{W}}}\|\mathbf{X} - \hat{\mathbf{W}}\hat{\mathbf{W}}^\mathsf{T}\mathbf{X}\|_F^2 \\
&= \arg\min_{\hat{\mathbf{W}}} \operatorname{tr}\left((\mathbf{X} - \hat{\mathbf{W}}\hat{\mathbf{W}}^\mathsf{T}\mathbf{X})^\mathsf{T}(\mathbf{X} - \hat{\mathbf{W}}\hat{\mathbf{W}}^\mathsf{T}\mathbf{X})\right) \\
&= \arg\min_{\hat{\mathbf{W}}} \operatorname{tr}\left(\mathbf{X}^\mathsf{T}\mathbf{X}\right) - \operatorname{tr}\left(\hat{\mathbf{W}}^\mathsf{T}\mathbf{X}\mathbf{X}^\mathsf{T}\hat{\mathbf{W}}\right) \\
&= \arg\max_{\hat{\mathbf{W}}} \operatorname{tr}\left(\hat{\mathbf{W}}^\mathsf{T}\mathbf{X}\mathbf{X}^\mathsf{T}\hat{\mathbf{W}}\right)
\end{aligned}$$

subject to $\hat{\mathbf{W}}^\mathsf{T}\hat{\mathbf{W}} = \mathbf{I}$ which is equivalent to the optimization problem (**??**).

5 Link between SVD and PCA

We investigate the relationship between PCA and SVD on a set of centralized data stored in matrix $\mathbf{X}$. The projection basis of PCA is given by the r eigenvectors of the covariance matrix:

$$\mathbf{X}\mathbf{X}^\mathsf{T} = \mathbf{W}_r\Lambda_r\mathbf{W}_r^\mathsf{T}. \tag{8}$$

Assume that the SVD decomposition of

$$\mathbf{X} = \mathbf{U}_r\Sigma_r\mathbf{V}_r^\mathsf{T}.$$

Therefore, the covariance matrix can be written as

$$\mathbf{X}\mathbf{X}^\mathsf{T} = \mathbf{U}_r\Sigma_r\mathbf{V}_r^\mathsf{T}\mathbf{V}_r\Sigma_r^\mathsf{T}\mathbf{U}_r^\mathsf{T} = \mathbf{U}_r\Sigma_r^2\mathbf{U}_r^\mathsf{T}.$$

Comparing to (8), we find that $\mathbf{W}_r = \mathbf{U}_r$ and $\mathbf{\Lambda}_r = \Sigma_r^2$.