Projecting onto a line

Consider the standard setting where we are given \( n \) points in \( d \) dimensions. Call these \( \vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n \). As before, our goal is to reduce the number of dimensions to a small number \( k \). In principal component analysis (or PCA), we will model the data by a \( k \)-dimensional subspace, and find the subspace for which the error in this representation is smallest.

Suppose \( k = 1 \). Then we want to approximate the data with a line. Assume the data is centered, so that \( \sum_{i=1}^{n} \vec{x}_i = 0 \), and that the line passes through the origin. Let the line correspond to direction \( \vec{w} \), a unit vector. What is the error in approximating \( \vec{x}_i \) with \( \vec{w} \)? We can use the perpendicular distance between the point \( \vec{x}_i \) and the line represented by \( \vec{w} \). It is easy to check that the perpendicular is given by \( \vec{x}_i - (\vec{x}_i \cdot \vec{w}) \vec{w} \), so that its squared length is

Squared error in \( \vec{x}_i = \| \vec{x}_i - (\vec{x}_i \cdot \vec{w}) \vec{w} \|^2 = \| \vec{x}_i \|^2 - 2(\vec{x}_i \cdot \vec{w})^2 + (\vec{x}_i \cdot \vec{w})^2 \times \| \vec{w} \|^2 = \| \vec{x}_i \|^2 - (\vec{x}_i \cdot \vec{w})^2

where the final equality uses that \( \vec{w} \) is a unit vector. Summing this over all the points, we have:

\[
MSE(\vec{w}) = \frac{1}{n} \left[ \sum_{i=1}^{n} \| \vec{x}_i \|^2 - \sum_{i=1}^{n} (\vec{x}_i \cdot \vec{w})^2 \right]
\]

We want to choose \( \vec{w} \) to minimize this error. Since the first term is independent of \( \vec{w} \), this corresponds to maximizing

\[
\sigma^2 = \frac{1}{n} \sum_{i=1}^{n} (\vec{x}_i \cdot \vec{w})^2
\]

Centering. We assumed above that the data has mean \( \bar{0} \) and that the line \( \vec{w} \) passes through the origin. If the data is not centered, one can show that the line with minimum error will pass through the mean \( \bar{x} = \frac{1}{n} \sum_{i=1}^{n} \vec{x}_i \). To see this, suppose we translate the points by \( \vec{y} \) and then take the optimal line that passes through the origin. The problem then corresponds to minimizing over \( \vec{y} \) and unit vector \( \vec{w} \) the quantity:

\[
MSE(\vec{w}, \vec{y}) = \frac{1}{n} \left[ \sum_{i=1}^{n} \| \vec{x}_i - \vec{y} \|^2 - \sum_{i=1}^{n} ((\vec{x}_i - \vec{y}) \cdot \vec{w})^2 \right]
\]

Taking the partial derivative with respect to \( \vec{y} \) gives

\[
\frac{\partial MSE(\vec{w}, \vec{y})}{\partial \vec{y}} = -2(\bar{x} - \vec{y} - ((\bar{x} - \vec{y}) \cdot \vec{w}) \vec{w})
\]

Since this is zero, we get that \( \bar{x} - \vec{y} \) is parallel to \( \vec{w} \), so that \( \vec{y} = \bar{x} + c\vec{w} \). Plugging this into the MSE formula, it is easy to check that all these values of \( \vec{y} \) give the same MSE, so without loss of generality, we can set \( \vec{y} = \bar{x} \), and center the data at the mean.
PROJECTING ONTO A LINE

Computing the Solution. How do we find this direction \( \vec{w} \)? For this, we resort to matrix notation. Let \( X \) be a \( n \times d \) matrix where row \( i \) is the vector \( \vec{x}_i \). Then, the lengths of the projections of the points onto direction \( \vec{w} \) is given by the vector \( X\vec{w} \). We have:

\[
\sigma^2 = \frac{1}{n} \sum_{i=1}^{n} (\vec{x}_i \cdot \vec{w})^2 = \frac{1}{n} (X\vec{w})^T (X\vec{w}) = \vec{w}^T \frac{X^T X}{n} \vec{w}
\]

The goal is to find unit vector \( \vec{w} \) to maximize this quantity. But \( \vec{w} \) is precisely the first eigenvector of \( V = \frac{X^T X}{n} \).

What is the matrix \( V \)? The \((i,j)\)th entry of \( V \) is given by

\[
V_{ij} = \frac{1}{n} \sum_{r=1}^{n} x_{ri} x_{rj}
\]

where \( x_{ri} \) is the \( i \)th dimension of \( \vec{x}_r \). The above quantity is the covariance between dimensions \( i \) and \( j \), so that \( V \) is the \( d \times d \) matrix of covariances.

Intuition. As an example, suppose the data is the two-dimensional ellipse \( x^2 + y^2 = 1 \), where \( \lambda > 1 \). The covariance between \( x \) and \( y \) is zero, since the data is symmetric around both axes. Further, the covariance between \( x \) and itself is \( \lambda \), and between \( y \) and itself is 1. Therefore \( V \) is the diagonal matrix, where the first row is \((2,0)\) and the second row is \((0,1)\). The map \( \vec{w} \to V\vec{w} \) maps points on a circle to points on the ellipse. This means the first eigenvector, the direction of maximum stretch, is the \( x \) axis, and indeed, this is the direction where the data has greatest variance.

In general, if the data lies on an ellipse, the covariance matrix \( V \) can be written as \( V = QDQ^T \). It takes points on a circle, rotates them by \( Q^T \), stretches them by \( D \), and rotates back by \( Q \). The direction of maximum stretch is the first eigenvector of \( V \), and gives the first principal component.

A different way to get intuition is to note that \( V \) is a \( d \times d \) matrix, and once you have \( V \), you can throw the data away. If we are just given the mean \( \vec{\mu} \) and the covariance matrix \( \Sigma \), we can pretend the data is generated by a “maximum entropy” distribution with these parameters. This is the multidimensional normal distribution \( \mathcal{N}(\vec{\mu}, \Sigma) \). Given \( \vec{x} \sim \mathcal{N}(\vec{\mu}, \Sigma) \), we have

\[
\vec{\mu} = \mathbf{E}[\vec{x}] \quad \text{and} \quad \Sigma = \mathbf{E} \left[ (\vec{x} - \vec{\mu})(\vec{x} - \vec{\mu})^T \right]
\]

This distribution is centered at \( \vec{\mu} \). Since we subtracted \( \vec{\mu} \) from all the data points, we assumed \( \vec{\mu} = \vec{0} \). The density function is given by

\[
 f(\vec{x}) = \frac{1}{\sqrt{(2\pi)^d|\Sigma|}} e^{-\frac{1}{2} \vec{x}^T \Sigma^{-1} \vec{x}}
\]

Note that for \( d = 1 \), \( \Sigma \) is the variance of the normal distribution, and the above formula is just the density of \( \mathcal{N}(0, \sigma^2) \).

The normal distribution is symmetric about an ellipse, but this ellipse is not aligned with the coordinate axes. The eigenvectors of \( \Sigma \) are the directions corresponding to the axes, and the corresponding eigenvalues are how much the ellipse stretches in each direction (that is, the corresponding variance). The direction in which the normal distribution has highest variance is therefore the first eigenvector, and the corresponding variance is \( \lambda_1 \).
Higher dimensional approximations

The above approach can be generalized to ask what is the best \( k \)-dimensional subspace that approximates the \( n \) points. Let the \( k \) dimensional subspace be given by the orthonormal vectors \( \vec{w}_1, \vec{w}_2, \ldots, \vec{w}_k \). Then, the projection of \( \vec{x}_i \) onto this space is given by

\[
\sum_{j=1}^{k} (\vec{x}_i \cdot \vec{w}_j) \vec{w}_j
\]

We want to maximize the sum of the squares of the lengths of these projections, since that would minimize the sum of the squares of the perpendicular distances. Therefore, we wish to solve

\[
\text{Maximize} \quad \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{k} (\vec{x}_i \cdot \vec{w}_j)^2 = \sum_{j=1}^{k} \vec{w}_j^T V \vec{w}_j
\]

subject to

\[
\vec{w}_j^T \vec{w}_j = 1 \quad \forall j = 1, 2, \ldots, k
\]

We can use the method of Lagrange multipliers. Let \( \lambda_j \) denote the Lagrange multiplier for the constraint \( \vec{w}_j^T \vec{w}_j = 1 \). Let

\[
\mathcal{L}(\{\vec{w}_j\}, \vec{\lambda}) = \sum_{j=1}^{k} \left( \vec{w}_j^T V \vec{w}_j - \lambda_j (\vec{w}_j^T \vec{w}_j - 1) \right)
\]

The claim is that maximizing this quantity is the same as the original maximization. To see this, at the maximum point of \( \mathcal{L} \), we must have for all \( j \):

\[
\frac{\partial \mathcal{L}}{\partial \lambda_j} = 0 \quad \Rightarrow \quad \vec{w}_j^T \vec{w}_j = 1
\]

Therefore, at the optimum of \( \mathcal{L} \), the constraints in the original problem must be satisfied. Plugging the above equalities into the function \( \mathcal{L} \), the terms \( \lambda_j \) disappear and it turns into the original objective. Therefore, maximizing \( \mathcal{L} \) with respect to \( \lambda_j \) and \( \vec{w}_j \) is sufficient to solve the original optimization problem.

Differentiating \( \mathcal{L} \) with respect to each coordinate in \( \vec{w}_j \), we have

\[
\frac{\partial \mathcal{L}}{\partial \vec{w}_j} = 2V \vec{w}_j - 2\lambda_j \vec{w}_j = 0
\]

This means each \( \lambda_j \) is an eigenvalue of \( V \), and \( \vec{w}_j \) is the corresponding eigenvector. Then, \( \vec{w}_j^T V \vec{w}_j = \lambda_j \), so that the objective is \( \sum_{j=1}^{k} \lambda_j \). This is maximized by choosing the top \( k \) eigenvectors as the subspace to project onto, and the corresponding variance in the data is \( \sum_{j=1}^{k} \lambda_j \).

The above result is somewhat surprising – we started with a joint optimization problem of finding a \( k \)-dimensional subspace with minimum error. Turned out the algorithm for finding such a subspace is greedy – choose dimensions one at a time, that is, solve the above optimization problem one dimension at a time. It was a priori not obvious that a greedy algorithm that sequentially expands the basis would be optimal for this problem.
Computation

Note that since $V = X^T X$, this means $V$ is positive semidefinite, and has all non-negative eigenvalues. To compute the largest eigenvector, we can use power iteration. To compute the second eigenvector, we perform power iteration, but at each step, project the resulting vector onto the direction perpendicular to the first eigenvector. In the limit, this will converge to the second eigenvector’s direction, and so on.

Discussion

The first $k$ principal components give an approximation to the input data as

$$
\tilde{x}_i = \bar{x} + \sum_{j=1}^{k} c_{ij} \bar{w}_j
$$

We can change the basis to the principal components and attempt to interpret the coefficients $c_{ij}$ as well as these directions. This sometimes has value, but at other times, can be misleading. The main reason is that these directions are orthogonal, and further, the coefficients can be negative. Finally, this method is good when the data lies close to a linear subspace, but if the data lies on a circle, principal components may yield meaningless results.

Singular Value Decomposition

The SVD takes the method of principal components one step further. Instead of just finding a low dimensional representation of the dimensions (or features of the data), it also attempts to compress the data points themselves from $m$ down to a smaller number. The input is a $m \times n$ matrix whose rank is $r$.

**Definition 1.** The rank of a matrix $M$ is the largest number of rows (or columns) that are linearly independent. If the data is viewed as $m$ points in $n$ dimensions, this is the dimension of the linear subspace that contains the points.

The singular value decomposition is a way of viewing what a matrix does to a vector. Recall that a symmetric matrix can be viewed as rotating a vector, stretching it, and rotating it back, so that the directions corresponding to the top eigenvectors are stretched the most. Here is a similar view for any matrix. Suppose $M$ has rank $n$. Then, the matrix $M^T M$ is positive semidefinite with $n$ eigenvalues strictly greater than zero. Suppose

$$M^T M = V \Sigma^2 V^T$$

is the PCA of $M$ viewed as $n$ data points in $d$ dimensions. Therefore, $D$ is the diagonal matrix of eigenvalues of $M^T M$, $\Sigma$ is the diagonal with the square roots of these values, and $V$ is the matrix whose columns are the eigenvectors of $M^T M$. Suppose we set

$$U = MV \Sigma^{-1}$$

Then it is easy to check that since $\Sigma^{-1}$ is also a diagonal matrix, we have

$$U^T U = \Sigma^{-1} V^T M^T M V \Sigma^{-1} = \Sigma^{-1} \Sigma^2 \Sigma^{-1} = I$$
Connection to PCA

What does this mean? Suppose we project the rows (data points) of $M$ onto the first principal component $v_1$, then the lengths of the projections written as a $m$-dimensional vector are $\sigma_1 u_1$. Similarly for the projections onto $v_2$ and so on. The interesting point is that these projected lengths, when themselves viewed as vectors, are orthogonal, which is a surprising observation!

The only catch is that $U$ is not a square matrix since its dimensions is $m \times n$, and further $\Sigma$ may not be invertible if there are zero eigenvalues. However, we can extend $U$ to an orthonormal basis to get a $m \times m$ matrix, and we get the following theorem that we state without proof.

**Theorem 1.** Suppose $M$ is a $m \times n$ matrix with real entries. Then there exist orthonormal matrices $U_{m \times m}$ and $V_{n \times n}$, and a diagonal matrix $\Sigma_{m \times n}$, all with real entries, so that:

- $\Sigma = \text{diagonal}(\sigma_1, \sigma_2, \ldots, \sigma_p)$ where $p = \min(m, n)$ with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0$, and
- $MV = U\Sigma$, that is, $M = U\Sigma V^T$.

If $M$ has rank $r$, then $\sigma_r > \sigma_{r+1} = \sigma_{r+2} = \cdots = \sigma_p = 0$. In this case,

$$M = \sum_{k=1}^r \sigma_k u_k v_k^T$$

What does this mean? Suppose we view the columns of $V$ as $r$ vectors on a unit sphere in $n$ dimensions. Then, $M$ maps $v_k$ to $\sigma_k u_k$, that is, to points on an ellipse in $m$ dimensions that is stretched by $\sigma_k$ in orthogonal directions $u_k$. (These directions will be the principal axes of the ellipse.) The quantities along the diagonal of $\Sigma - \sigma_1, \sigma_2, \ldots, \sigma_r$ – are termed the singular values of $M$. The columns of $U$ are the left singular vectors, and those of $V$ are termed the right singular vectors. Note that the range of $M$ is given by the span of the vectors $u_1, u_2, \ldots, u_r$, and the null-space of $M$, that is the set of vectors $x$ such that $Mx = 0$ is the span of $v_{r+1}, v_{r+2}, \ldots, v_n$.

Further, note that for each $k = 1, 2, \ldots, r$, the quantity $u_k v_k^T$ is a $m \times n$ matrix of rank 1 since each row of this matrix is the vector $v_k^T$ multiplied by a component of $u_k$. This means $M$ is a sum of $r$ matrices of rank 1.

**Connection to PCA**

By the definition of PCA, $V$ define the eigenvectors of the symmetric matrix $M^T M$, and the eigenvalues are $\sigma_1^2, \sigma_2^2, \ldots$. The matrix $V$ are termed the right singular vectors. Similarly, one can check that $U$ define the eigenvectors of $MM^T$. These are termed the left singular vectors. The SVD gives an interpretation for both the data points and the dimensions in terms of rank 1 concepts, while the PCA short-cuts this and gives an interpretation for just the data points (rows). It also gives a way to compute the SVD by computing the eigenvectors of $MM^T$ and $M^T M$. It is easy to check that the eigenvalues of $MM^T$ and $M^T M$ are the same, and are given by $\sigma_1^2, \sigma_2^2, \ldots$. The eigenvectors are different, and are given by $U$ and $V$ respectively.

Since the SVD generalizes the PCA, we can interpret the SVD as providing lower dimensional approximations to $M$. For this, we first define the Frobenius norm of a matrix.

**Definition 2.** The Frobenius norm of a matrix $M$ is defined as

$$\|M\|_F = \sqrt{\sum_{i,j} m_{ij}^2}$$
Note that the square of the Frobenius norm is the sum of the squared lengths of the rows (resp. columns) of $M$. Since $U^T M V = \Sigma$, this operation corresponds to rotating the rows and then columns of $M$, both of which preserve the lengths of the rows (resp. columns). This means the Frobenius norm of $M$ is the same as that of $\Sigma$, which is $\sum_{k=1}^{r} \sigma_k^2$. Further, the matrix induced by the top $k$ terms is the matrix of rank $k$ that minimizes the root mean squared error with respect to $M$. In other words,

**Theorem 2.** Given any $m \times n$ matrix $M$ of rank $r$, for any $k \leq r$, the matrix $M_k = \sum_{s=1}^{k} \sigma_s u_s v_s^T$ minimizes $\|M - B\|_F$ over all matrices $B$ of rank $k$. The minimum value is simply $\sum_{s=k+1}^{r} \sigma_s^2$.

When $k = 1$, consider the matrix $M_1 = \sigma_1 u_1 v_1^T$. The rows are all scalar multiples of each other, so that the direction $v_1^T$ is the canonical direction for interpreting the rows. Similarly, the columns are scalar multiples of each other, and $u_1$ is the canonical direction to interpret columns. This defines a **concept**, and the weight of this concept is given by $\sigma_1$. Note that minimizing the Frobenius norm with respect to a rank 1 matrix is exactly the same as asking for the line which minimizes the sum of squares of distances of the $m$ points to this line, and the same for higher dimensions. This gives an immediate proof of the above theorem via the definition of PCA.

There are several applications of SVD, collectively termed **topic modeling**. For instance, when the matrix is documents versus phrases contained in them, the concepts discovered will be **topics** in the documents; if the matrix is genes versus tissues in which these genes are expressed, the concepts could be co-expressed pathways of genes that arise in several tissues. In such cases, we need the concepts to have positive entries, else they are very hard to interpret. This leads to the area of **non-negative matrix factorization**. In its full generality, finding such a factorization of rank $k$ that minimizes the Frobenius norm of the error is NP-HARD. However, several heuristics, most notably Latent Dirichlet Allocations (LDA) are widely used to solve the topic modeling problem in practice.

**Least Squares Regression and the Pseudoinverse**

A simple application of the SVD is in computing the least squares solution to a set of linear equations. Suppose we want to solve $A\bar{x} = \bar{b}$ for $\bar{x}$, but there is no feasible solution. In that case, a reasonable thing to do is to find that $\bar{x}$ that minimizes the norm $\|A\bar{x} - \bar{b}\|$. A different application is when the rows of $A$ are $m$ observed data points in $n$ dimensions, and $\bar{b}$ is the independent variable. So the $i^{th}$ data point is $\bar{a}_i$, and the corresponding outcome is $b_i$. The goal is to explain the independent variable as a linear function of the observed data, that is find a linear function $\bar{x}$ so that $\bar{b}_i$ is as close as possible to $\sum_{j=1}^{n} a_{ij} x_j$, for all data points $i$. In order to do this, we minimize the sum of the squared errors between $b_i$ and the linear prediction, which is minimizing $\|A\bar{x} - \bar{b}\|$. This is termed the **least squared regression** problem in machine learning.

Note that in the PCA, where there is no independent variable and the goal is to find a linear function which minimizes **perpendicular** distances to this line. In the least squares problem, the goal is to minimize the **vertical** distance from the independent variable to the linear prediction. Further, we could solve the least squares problem by various means, including brute force differentiation. The algorithm below is simply more elegant than any other approach.

The SVD gives a simple algorithm for computing the least squares solution. If the SVD of $A$ is $U\Sigma V^T$, least squares corresponds to minimizing

$$\|U\Sigma V^T \bar{x} - \bar{b}\| = \|U \left( \Sigma V^T \bar{x} - U^T \bar{b} \right)\|$$
Note now that multiplication by the orthonormal matrix $U$ preserves the norm. Further, we can set $\vec{y} = V^T \vec{x}$, and $\vec{c} = U^T \vec{b}$, so that we want to minimize

$$\| \Sigma \vec{y} - \vec{c} \|$$

This is a simpler problem: The quantity $\Sigma \vec{y}$ is a vector whose $k^{th}$ component for $k \in \{1, 2, \ldots, r\}$ is $\sigma_k y_k$, and the remaining components are zero. This means to minimize the norm, we simply set $y_k = \frac{c_k}{\sigma_k}$ for $k = 1, 2, \ldots, r$, and the remaining $y_k$ to zero. Therefore, we can write $\vec{y} = \Sigma^+ \vec{c}$, where $\Sigma^+$ is a $n \times m$ diagonal matrix whose $k^{th}$ term for $k = 1, 2, \ldots, r$ is $\frac{1}{\sigma_k}$.

Therefore, the least squares solution to $A \vec{x} = \vec{b}$, where the SVD of $A$ is given by $A = U \Sigma V^T$, is:

$$\vec{x} = V \vec{y} = V \Sigma^+ \vec{c} = V \Sigma^+ U^T \vec{b}$$

The quantity $A^* = V \Sigma^+ U^T$ is termed the Moore-Penrose pseudoinverse of $A$. Note that unlike the standard inverse, the pseudoinverse always exists. This gives a closed form for the least squares solution in terms of the SVD of $A$. 