CS 189 Introduction to Machine Learning
Fall 2017 Note 9

# 1 Principal Component Analysis

In machine learning, the data we have are often very high-dimensional. There are a number of reasons why we might want to work with a lower-dimensional representation:

- Visualization (if we can get it down to 2 or 3 dimensions), e.g. for exploratory data analysis
- Reduce computational load
- Reduce noise

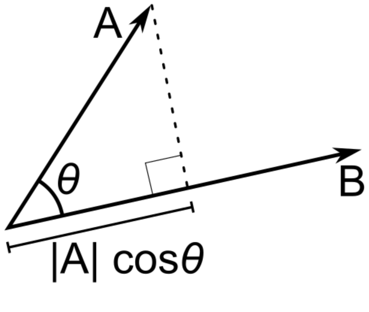
**Principal Component Analysis (PCA)** is an unsupervised dimensionality reduction technique. Given a matrix of data points, it finds one or more orthogonal directions that capture the largest amount of variance in the data. Intuitively, the directions with less variance contain less information and may be discarded without introducing too much error.

## 1.1 Projection

Let us first review the meaning of scalar projection of one vector onto another. If $u \in \mathbb{R}^d$ is a unit vector, i.e. $\|u\| = 1$, then the projection of another vector $x \in \mathbb{R}^d$ onto $u$ is given by $x^\top u$. This quantity tells us roughly how much of the projected vector $x$ lies along the direction given direction $u$. Why does this expression make sense? Recall the slightly more general formula which holds for vectors of any length:

$$x^\top u = \|x\|\|u\|\cos\theta$$

where $\theta$ is the angle between the vectors. In this case, since $\|u\| = 1$, the expression simplifies to $x^\top u = \|x\|\cos\theta$. But since cosine gives the ratio of the adjacent side (the projection we want to find) to the hypotenuse ($\|x\|$), this is exactly what we want:



## 1.2 Formulating PCA

Let $X \in \mathbb{R}^{n \times d}$ be our matrix of data, where each row is a $d$-dimensional data point. We will assume that the data points have mean zero; otherwise we subtract the mean to make them zero-mean:

$$X - \frac{1}{n}\vec{1}_n\vec{1}_n^\top X, \text{ where } \vec{1}_n = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$$

The motivation for this is that we want to find directions of high variance within the data, and variance is defined relative to the mean of the data. If we did not zero-center the data, the directions found would be heavily influenced by where the data lie relative to the origin, rather than where they lie relative to the other data, which is more useful. For example, translating all the data by some fixed vector could completely change the principal components if we did not center.

Recall that for any random variable $Z$,

$$\text{var}(Z) = \mathbb{E}[(Z - \mathbb{E}[Z])^2]$$

so if $\mathbb{E}[Z] = 0$ then $\text{var}(Z) = \mathbb{E}[Z^2]$.

Hence once $X$ is zero-mean, the variance of the projections is given by

$$\varepsilon_{PCA\_var}(u) = \sum_{i=1}^{n}(x_i^\top u)^2 = \|Xu\|^2$$

where $u$ is constrained to have unit norm. We want to maximize the variance, so the objective becomes

$$\max_{\|u\|=1} \varepsilon_{PCA\_var}(u) = \max_{\|u\|=1} \|Xu\|^2 = \max_{u \neq 0} \frac{\|Xu\|^2}{\|u\|^2} = \max_{u \neq 0} \frac{u^\top X^\top X u}{u^\top u}$$

The ratio on the right is known as a *Rayleigh quotient*. We will see that Rayleigh quotients are heavily related to eigenvalues, so anytime you see one, your eigensense should tingle.

## 1.3 Rayleigh Quotients

Suppose $M \in \mathbb{R}^{d \times d}$ is a real, symmetric ($M = M^\top$) matrix. The Rayleigh quotient of $M$ is defined as

$$R(u; M) = \frac{u^\top M u}{u^\top u}$$

Denote the eigenvalues of $M$ by $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$, with corresponding eigenvectors $v_1, \ldots, v_d$. That is, $Mv_j = \lambda_j v_j$ for $j = 1, \cdots, d$. If we stack the $v_j$ as columns of a matrix $V$:

$$V = \begin{bmatrix} v_1 & \cdots & v_d \end{bmatrix}$$

then the eigenvector equations can be simulatneously written as

$$MV = V\Lambda$$

where

$$\Lambda = \text{diag}(\lambda_1, \cdots, \lambda_d)$$

Then since $V$ is an orthogonal matrix, it is invertible with $V^{-1} = V^\top$, so

$$V^\top M V = \Lambda$$

Let $u$ be a unit length vector. Since $\{v_j\}$ form a basis, we can write

$$u = \sum_{j=1}^{d} \alpha_j v_j = V\alpha$$

Then since $V$ is an orthogonal matrix, $\|\alpha\| = 1$ as well. Now

$$R(u;M) = \frac{u^\top M u}{u^\top u} = \alpha^\top V^\top M V \alpha = \alpha^\top \Lambda \alpha = \sum_{j=1}^{d} \lambda_j \alpha_j^2$$

Since we have the constraint $\alpha_1^1 + \cdots + \alpha_d^2 = 1$, the right-most expression is a weighted average of the eigenvalues and hence bounded by the smallest and largest of these:

$$\lambda_d \leq R(u;M) \leq \lambda_1$$

The lower bound is achieved by putting $\alpha_d = \pm 1$, $\alpha_{j \neq d} = 0$, and the upper bound by $\alpha_1 = \pm 1$, $\alpha_{j \neq 1} = 0$. The maximizing value can then be recovered by

$$\sum_{j=1}^{d} \alpha_j v_j = \pm v_1$$

That is, it is an eigenvector corresponding to the largest eigenvalue! Hence

$$\lambda_d = R(v_d;M) \leq R(u;M) \leq R(v_1;M) = \lambda_1$$

Finally, note that since the Rayleigh quotient is scale invariant, i.e. $R(\gamma u;M) = R(u;M)$ for any $\gamma \neq 0$, the inequality above holds for any scaling of the vectors, not just unit-length vectors.

## 1.4 Calculating the first principal component

Armed with our knowledge of Rayleigh quotients, the solution to the PCA problem is immediate:

$$\max_{\|u\|=1} \varepsilon_{PCA\_var}(u) = \max_{u \neq 0} \underbrace{\frac{u^\top X^\top X u}{u^\top u}}_{R(u,X^\top X)} = \lambda_1(X^\top X)$$

where the maximizer $u^*$ is a unit eigenvector corresponding to this eigenvalue. Writing $X = U\Sigma V^\top$, we have

$$X^\top X = V\Sigma^\top \underbrace{U^\top U}_{I} \Sigma V^\top = V\Sigma^\top \Sigma V^\top$$

The expression on the right is an eigendecomposition of $X^\top X$, so

$$\lambda_1(X^\top X) = [\Sigma^\top \Sigma]_{11} = \sigma_1^2(X)$$

with corresponding eigenvector $\vec{v}_1$, which is the first principal component.

## 1.5 Finding multiple principal components

We have seen how to derive the first principal component, which maximizes the variance of the projected data. But usually we will need more than one direction, since one direction is unlikely to capture the data well. The basic idea here is to subtract off the contributions of the previously computed principal components, and then apply the same rule as before to what remains. If $u_{(1)}, \ldots, u_{(k-1)}$ denote the principal components already computed, this subtracting off is accomplished by

$$\hat{X} = X - \sum_{j=1}^{k-1} X u_{(j)} u_{(j)}^\top = X \left( I - \sum_{j=1}^{k-1} u_{(j)} u_{(j)}^\top \right)$$

This expression should be understood as applying the same subtracting transformation to each row of the data[1]:

$$\hat{x}_i = \left( I - \sum_{j=1}^{k-1} u_{(j)} u_{(j)}^\top \right) x_i = x_i - \sum_{j=1}^{k-1} u_{(j)} u_{(j)}^\top x_i$$

The vector $u_{(j)} u_{(j)}^\top x_i$ should be recognized as the orthogonal projection of $x_i$ onto the subspace spanned by $u_{(j)}$. Hence $\hat{x}_i$ is what's left when you start with $x_i$ and then remove all the components that belong to the subspaces spanned by each $u_{(j)}$.[2]

We want to find the direction of largest variance subject to the constraint that it must be orthogonal to all the previously computed directions. Thus we have a constrained problem of the form

$$u_{(k)} = \underset{\forall j<k:\ u^\top u_{(j)}=0}{\arg\max} \frac{u^\top \hat{X}^\top \hat{X} u}{u^\top u}$$

But we don't want to actually compute $\hat{X}$. Fortunately, we don't have to! Consider that if $u$ is orthogonal to $u_{(1)}, \ldots, u_{(k-1)}$ (as we constrain it to be), then

$$\hat{X} u = \left( X - \sum_{j=1}^{k-1} X u_{(j)} u_{(j)}^\top \right) u = X u - \sum_{j=1}^{k-1} X u_{(j)} \underbrace{u_{(j)}^\top u}_{0} = X u$$

Thus we can write the optimization problem above as

$$u_{(k)} = \underset{\forall j<k:\ u^\top u_{(j)}=0}{\arg\max} \frac{u^\top X^\top X u}{u^\top u}$$

eliminating the need to compute $\hat{X}$. Unsurprisingly, the solution to this problem is given by $u_{(k)} = v_k$, that is, a unit eigenvector corresponding to the $k$th largest eigenvalue of $X^\top X$.

Rather than iteratively computing each new $u_{(j)}$, we can view the problem of finding the first $k$ principal components as a joint optimization problem over all $k$ directions simultaneously. This

[1] To see this, take the transpose of both sides and use the symmetry of $I - \sum_j u_{(j)} u_{(j)}^\top$.

[2] This is exactly the same idea as the Gram-Schmidt process.

amounts to maximizing the variance as projected onto a $k$-dimensional subspace:

$$U = \underset{U^\top U=I}{\arg\max} \sum_{j=1}^{k} u_{(j)}^\top X^\top X u_{(j)} = \underset{U^\top U=I}{\arg\max}\, \mathrm{tr}(U^\top X^\top X U)$$

For matrices $U$ with orthogonal columns we can define

$$R(U;M) = R([u_1,\dots,u_k];M) = \sum_{j=1}^{k} R(u_j;M)$$

As before, the bounds for this expression are given in terms of the smallest and largest eigenvalues, but now there are $k$ of them:

$$\begin{aligned}\lambda_1+\lambda_2+\cdots+\lambda_k &= R([v_1,v_2,\dots,v_k];M)\\ &\geq R(U;M)\\ &\geq R([v_{d-k+1},\dots,v_{d-1},v_d];M)\\ &= \lambda_{d-k+1}+\cdots+\lambda_{d-1}+\lambda_d\end{aligned}$$

Hence, projection onto the subspace spanned by the first $k$ leading eigenvectors maximizes the variance of the projected data. We can find $k$ principal components by computing the SVD, $X = U\Sigma V^\top$, and then taking the first $k$ columns of the matrix $V$.

## 1.6 Projecting onto the PCA coordinate system

Once we have the principal components, we can use them as a new coordinate system. To do this we must project the data onto this coordinate system, which can be done in the same way as above (taking inner products). Each data point $x_i \in \mathbb{R}^d$ becomes a new vector $\tilde{x}_i \in \mathbb{R}^k$, where $k$ is the number of principal components. The components of the projection write

$$[\tilde{x}_i]_j = x_i^\top u_j$$

We can compute all these vectors at once more efficiently using a matrix-matrix multiplication

$$\tilde{X} = XU$$

where $U \in \mathbb{R}^{d\times k}$ is a matrix whose columns are the principal components.

Below we plot the result of such a projection in the case $d = k = 2$:

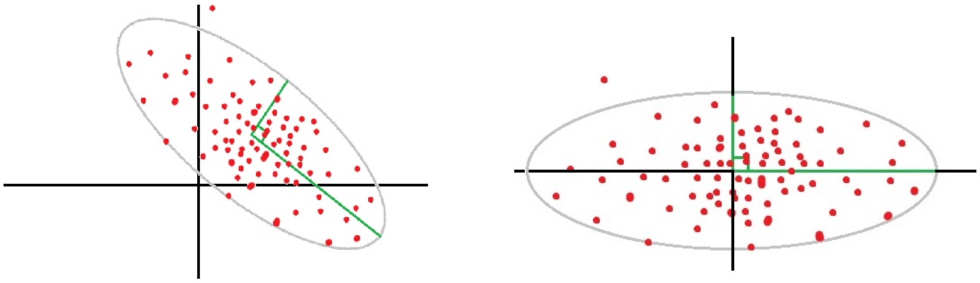


Figure 1: Left:data points; Right: PCA projection of data points

Observe that the data are uncorrelated in the projected space. Also note that this example does not show the full power of PCA since we have not reduced the dimensionality of the data at all – the plot is merely to show the PCA coordinate transformation.

# 2 Other derivations of PCA

We have given the most common derivation of PCA above, but it turns out that there are other equivalent ways to arrive at the same formulation. These give us helpful additional perspectives on what PCA is doing.

## 2.1 Gaussian assumption

Let us assume that the data are generated by a multivariate Gaussian distribution:

$$x_i \overset{\text{i.i.d.}}{\sim} \mathscr{N}(\mu, \Sigma)$$

Then the maximum likelihood estimate of the covariance matrix $\Sigma$ is

$$\hat{\Sigma} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(x_i - \bar{x})^\top = \frac{1}{n}X^\top X$$

where $\bar{x}$ is the sample average and the matrix $X$ is assumed to be zero-mean as before. The eigenvectors of $\hat{\Sigma}$ and $X^\top X$ are the same since they are positive scalar multiples of each other.

The contours of the multivariate Gaussian density form ellipsoids (see Figure 1). The direction of largest variance (i.e. the first principal component) is the eigenvector corresponding to the smallest eigenvalue of $\Sigma^{-1}$, which is the largest eigenvalue of $\Sigma$. We do not know $\Sigma$ in general, so we use $\hat{\Sigma}$ in its place. Thus the principal component is an eigenvector corresponding to the largest eigenvalue of $\hat{\Sigma}$. As mentioned earlier, this matrix has the same eigenvalues and eigenvectors as $X^\top X$, so we arrive at the same solution.

## 2.2 Minimizing reconstruction error

Ordinary least squares minimizes the vertical distance between the fitted line and the data points:

$$\|y - Xu\|^2$$

We show that PCA can be interpreted as minimizing the perpendicular distance between the principal component subspace and the data points, so in this sense it is doing the same thing as total least squares.

The orthogonal projection of a vector $x$ onto the subspace spanned by a unit vector $u$ equals $u$ scaled by the scalar projection of $x$ onto $u$:

$$P_u x = (uu^\top)x = (x^\top u)u$$

Suppose we want to minimize the total reconstruction error:

$$\begin{aligned}
\varepsilon_{PCA\_Err}(u) &= \sum_{i=1}^{n} \|x_i - P_u x_i\|^2 \\
&= \sum_{i=1}^{n} \left(\|x_i\|^2 - \|P_u x_i\|^2\right) && (*) \\
&= \sum_{i=1}^{n} \|x_i\|^2 - \sum_{i=1}^{n} \|(x_i^\top u)u\|^2 \\
&= \sum_{i=1}^{n} \|x_i\|^2 - \underbrace{\sum_{i=1}^{n} (x_i^\top u)^2}_{\varepsilon_{PCA\_Var}(u)}
\end{aligned}$$

where $(*)$ holds by the Pythagorean theorem

$$\|x - P_u x\|^2 + \|P_u x\|^2 = \|x\|^2$$

since $x - P_u x \perp P_u x$. Then since the first term $\sum_i \|x_i\|^2$ is constant with respect to $u$, we have

$$\arg\min_u \varepsilon_{PCA\_Err}(u) = \arg\min_u \ \text{constant} - \varepsilon_{PCA\_Var}(u) = \arg\max_u \varepsilon_{PCA\_Var}(u)$$

Hence minimizing reconstruction error is equivalent to maximizing projected variance.