

Linear Dimensionality Reduction

Often we want to find a representation of data from \mathbb{R}^D in some lower-dimensional space, \mathbb{R}^K , for $K \ll D$. For $K \in \{2, 3\}$, this is useful for visualization. For other K , it's useful if we believe that the data is noisy, or not ideal for our learning algorithm (eg., k NN).

Have data matrix in $\mathbf{X} \in \mathbb{R}^{N \times D}$. Want to linearly *project* \mathbf{X} into some $\mathbf{Z} \in \mathbb{R}^{N \times K}$. We don't want to lose much "information."

Two ways of deriving PCA:

- Project \mathbf{X} onto basis vectors of highest variance
- Project \mathbf{X} in such a way that the *reconstruction error* is minimized
- Imagine data was generated by an K -dimensional Gaussian and then noisified into D dimensions (wait until probabilistic models time)

(1) is standard, hence the name "principle component analysis" (a component is a basis vector).

First, center \mathbf{X} so it has mean 0.

Now, supposed we want to find a single dimension to project down on to. Represent this dimension by a vector \mathbf{u} , so that $\mathbf{Z} = \mathbf{X}\mathbf{u}$. We want to find \mathbf{u} so that $\mathbb{V}[\mathbf{Z}] = \mathbb{V}[\mathbf{X}\mathbf{u}]$ is maximized. We have to be careful to ensure that \mathbf{u} is a unit vector; otherwise we get trivial uninteresting solutions. This turns into an optimization problem:

$$\begin{aligned} \max_{\mathbf{u}: \|\mathbf{u}\|=1} \mathbb{V}[\mathbf{X}\mathbf{u}] &= \max_{\mathbf{u}: \|\mathbf{u}\|=1} \frac{1}{N} \sum_n (\mathbf{u}^\top \mathbf{x}_n)^2 \\ &= \max_{\mathbf{u}: \|\mathbf{u}\|=1} \frac{1}{N} \|\mathbf{X}\mathbf{u}\|^2 \end{aligned}$$

To solve this, we construct the Lagrangian by adding a multiplier for the constraint $\|\mathbf{u}\|^2 = 1$ (changing to squared norm doesn't change the problem); we also throw out the $\frac{1}{N}$ because it is a constant. Then we take the gradient and set it equal to zero.

$$\begin{aligned} \mathcal{L}(\mathbf{u}, \lambda) &= \|\mathbf{X}\mathbf{u}\|^2 - \lambda (\|\mathbf{u}\|^2 - 1) \\ \nabla_{\mathbf{u}} \mathcal{L} &= 2(\mathbf{X}^\top \mathbf{X})\mathbf{u} - 2\lambda \mathbf{u} = 0 \\ \iff (\mathbf{X}^\top \mathbf{X})\mathbf{u} &= \lambda \mathbf{u} \end{aligned}$$

So now all we need to do is solve the above for \mathbf{u} . Note that this has the form $\mathbf{A}\mathbf{u} = \lambda \mathbf{u}$ where \mathbf{A} is defined as $\mathbf{X}^\top \mathbf{X}$. The solution to this problem is a vector \mathbf{u} that is an *eigenvector* of \mathbf{A} and can be computed "easily" (use `eig` or `eigs` in Matlab/Octave).

When you get eigenvectors/eigenvalues of a matrix \mathbf{A} of size $D \times D$, you will get D -many pairs $(\mathbf{u}_i, \lambda_i)$. The λ s are measures of the variance along the corresponding \mathbf{u} , so we want to select the \mathbf{u}_i with largest corresponding λ_i .

In general, to project onto $K > 1$ dimensions, instead of choosing the single eigenvector with largest eigenvalue, we take the top K (they are guaranteed to be orthogonal).

The algorithm looks like:

- Center the data so \mathbf{X} has mean zero
- Compute $\mathbf{A} = \mathbf{X}^\top \mathbf{X}$
- Compute $\langle \mathbf{u}_i, \lambda_i \rangle_{i=1}^K$ the top K eigenvectors/values of \mathbf{A}
- Project $\mathbf{Z} = \mathbf{X}\mathbf{U}$, where \mathbf{U} is the $D \times K$ matrix comprised of the top K eigenvectors

An alternative way of thinking about PCA is as minimizing a reconstruction error. If we have a data point \mathbf{x}_n and project it with $\mathbf{x}_n \mathbf{U} \mapsto \mathbf{z}_n$, then we can “unproject” by $\mathbf{z}_n \mathbf{U}^\top \mapsto \tilde{\mathbf{x}}_n$. We can look at the reconstruction error $\|\mathbf{x}_n - \tilde{\mathbf{x}}_n\|^2$ and try to minimize this over \mathbf{U} , where we constrain \mathbf{U} to be orthonormal (i.e., orthogonal and unit vectors). The one-dimensional case is easier to see:

$$\begin{aligned} \min_{\mathbf{u}} \sum_n \|\mathbf{x}_n - \tilde{\mathbf{x}}_n\|^2 &= \min_{\mathbf{u}} \sum_n \|\mathbf{x}_n - \mathbf{x}_n \mathbf{u} \mathbf{u}^\top\|^2 \\ &= \min_{\mathbf{u}} \sum_n \left(\|\mathbf{x}_n\|^2 - (\mathbf{x}_n^\top \mathbf{u})^2 \right) \\ &= (\text{constant}) + \min_{\mathbf{u}} - \sum_n (\mathbf{x}_n^\top \mathbf{u})^2 \\ &= (\text{constant}) - \max_{\mathbf{u}} \sum_n (\mathbf{x}_n^\top \mathbf{u})^2 \\ &= (\text{constant}) - \max_{\mathbf{u}} \|\mathbf{X}\mathbf{u}\|^2 \end{aligned}$$

Here, the last line is exactly (modulo the constant) the same optimization we had in the variance case.

The *intuition* behind PCA is that we’re trying to find (non-axis-aligned) basis vectors along which \mathbf{X} has highest variance. Doing so turns into an eigenvalue problem which we know how to solve. The second intuition is that we’re trying to minimize reconstruction error (measured in terms of squared error) and this reverts to the same thing.