Mathematical Tools for Neuroscience (NEU 314)
Princeton University, Spring 2016
Jonathan Pillow

Lecture 9: PCA

1 Principal Components Analysis (PCA)

Suppose someone hands you a stack of N vectors, $\{\vec{x}_1, \ldots \vec{x}_N\}$, each of dimension d. For example, we might imagine we have made a simultaneous recording from d neurons, so each vector represents the spike counts of all recorded neurons in a single time bin, and we have N time bins total in the experiment.

We suspect that these vectors not "fill" out the entire d-dimensional space, but instead be confined to a lower-dimensional subspace. (For example, if two neurons always emit the same number of spikes, then their responses live entirely along the 1D subspace corresponding to the $x = y$ line).

Can we make a mathematically rigorous theory of dimensionality reduction that captures how much of the "variability" in the data is captured by a low-dimensional projection? (Yes: it turns out the tool we are looking for is PCA!)

1.1 Finding the best 1D subspace

.

Let's suppose we wish to find the best 1D subspace, i.e., the one-dimensional projection of the data that captures the largest amount of variability. We can formalize this as the problem of finding the unit vector $\vec{v}$ that maximizes the sum of squared linear projections of the data vectors:

$$\begin{aligned}\text{Sum of squared linear projections} = \sum_{i=1}^{N} (\vec{x}_i \cdot \vec{v})^2 &= ||X\vec{v}||^2 \\ &= (X\vec{v})^\top (X\vec{v}) \\ &= \vec{v}^\top X^\top X \vec{v} \\ &= \vec{v}^\top (X^\top X)\vec{v} \\ &= \vec{v}^\top C \vec{v},\end{aligned}$$

where $C = X^\top X$, under the constraint that $\vec{v}^\top \vec{v} = 1$.

It turns out that the solution corresponds to the top eigenvector of C (i.e., the eivenvector with the largest eigenvalue). We will prove this formally in two lectures time, but for now let's look at the SVD of the C matrix and look at what happens if we just restrict our choice of $\vec{v}$ to the eigenvectors of C.

First, remember that because C is symmetric and positive-semi-definite ("psd") its SVD is also its

eigenvector decomposition:

$$C = USU^\top,$$

where the columns of U are the eigenvectors and diagonal entries of S are eigenvalues.

Now let's consider what happens if we set $\vec{v} = \vec{u}_j$, i.e., the j'th eigenvector of C. Because U is an orthogonal matrix (i.e., its columns for an orthonormal basis), then $U^\top \vec{u}_j$ will be a vector of zeros with a 1 in the j'th component. We have

$$\begin{aligned}
\vec{u}_j^\top C \vec{u}_j &= \vec{u}_j^\top (USU^\top)\vec{u}_j \\
&= (\vec{u}_j U) S (U^\top \vec{u}_j) \\
&= \begin{bmatrix} 0 & \cdots & 1 & \cdots & 0 \end{bmatrix} \begin{bmatrix} s_1 & & & & \\ & \ddots & & & \\ & & s_j & & \\ & & & \ddots & \\ & & & & s_d \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \\
&= s_j
\end{aligned}$$

So: plugging in the j'th eigenvector of C gives us out s_j as the sum of squared projections. Since we want to maximize this quantity (i.e., find the linear projection that maximizes it), we should clearly choose the eigenvector with largest eigenvalue, which (since SVD orders them from greatest to smallest) corresponds to the solution

$$\vec{v} = \vec{u}_1$$

.

This vector (the "dominant" eigenvector of C) is the *first principle component* (sometimes called the "first PC" for short).

1.2 Finding best k-dimensional subspace

.

The above solution to the more general case of finding the k-dimensional subspace. Since the eigenvectors of C are orthogonal, the variability preserved by each projection is independent of the variability preserved by the other eigenvectors. Thus, a basis for the k-dimensional subspace that preserves the greatest variability is given by the first k eigenvectors of C:

$$\text{First } k \text{ PCs} = \begin{bmatrix} \vec{u}_1, & \ldots, & \vec{u}_k \end{bmatrix}$$

The total "variability" captured by these principal components is given by:

$$\begin{aligned}
\sum_{i=1}^{N} (\vec{x}_i \cdot \vec{u}_1)^2 + \cdots + \sum_{i=1}^{N} (\vec{x}_i \cdot \vec{u}_k)^2 &= ||X\vec{u}_1||^2 + \cdots + ||X\vec{u}_k||^2 \\
&= s_1 + \cdots + s_k
\end{aligned}$$

The variability of the *entire*x set of vectors is equal to

$$\sum_{i=1}^{N} ||\vec{x}_i||^2 = s_1 + \cdots + s_d.$$

(Verify for yourself that this is true!)

The *fraction* of the variability captured by the first k principal components is therefore given by

$$\frac{s_1 + \cdots + s_k}{s_1 + \cdots + s_k + \cdots + s_d}.$$

subsectionZero-centered data

PCA is tantamount to fitting an ellipse to your data: the eigenvectors $\vec{u}_i$ give the dominant axes of the ellipse, while the s_i gives the elongation of the ellipse along each axis, and is equal sum of squared projections (what we've been calling "variability" above) of the data along that axis.

So far we've assumed we wanted to maximize the sum of squared projections of the vector $\{\vec{x}_j\}$, which is equivalent to fitting an ellipse centered at zero and examining major axes of that ellipse. In many settings, however, we would prefer to examine the dimensionality of the data by considering an ellipse centered on the centroid (the arithmetic mean) of the data. We can achieve this by simply subtracting off the mean of the data from each data vector.

The mean is given by

$$\bar{x} = \frac{1}{N} \sum \vec{x}_i$$

.

We can call define the centered data matrix as

$$\tilde{X} = \begin{bmatrix} — & \vec{x}_1 & — \\ & \vdots & \\ — & \vec{x}_N & — \end{bmatrix} - \begin{bmatrix} — & \bar{x} & — \\ & \vdots & \\ — & \bar{x} & — \end{bmatrix},$$

that is, a matrix in which we subtract off the mean vector $\bar{x}$ from each row. Then by taking the eigenvectors of $(\tilde{X}^\top \tilde{X})$ we will be obtaining principal components of the centered data.

Note: this is the *standard* definition of PCA! It is uncommon to do PCA on uncentered data.

1.3 Notes on Matlab implementation

- We can center the data simply in matlab (i.e., without using a `for` loop) via the command `Xctr = X - repmat(mean(X),N,1);`.
- PCA is obtained by two additional lines:
  ```
  [U,S] = svd(Xctr'*Xctr);
  PCs_1toK = U(:,1:k);
  ```

- An even simpler method is to use `cov`, which computes the sample covariance matrix of the data. This centers the data and divides the $\tilde{X}^\top \tilde{X}$ matrix by N (or $N-1$, which is the default in Matlab). The resulting code for PCA is simply:
```
[U,S] = svd(cov(X));
PCs_1toK = U(:,1:k);
```

- We can get the variance captured by each PC in a single vector using:
```
s_vec = diag(S(1:k,1:k));
```