Chapter 6

Principal components

1. Introduction

The idea behind the principal components is roughly the following. Suppose one has a random vector $\mathbf{x} \in \mathbb{R}^d$ (or just a finite number of vectors $\mathbf{x}_j \in \mathbb{R}^d$, $j = 1, \ldots, n$ each chosen with the same probability $\frac{1}{n}$) and that the mean of these vectors is 0. Suppose that and one wants to choose project these vectors on a subspace with smaller dimension e.g. $m < d$) without loosing too much information. This is here taken to mean that if P is the projection, then the variance of $P\mathbf{x}$ is as large as possible, which turns out to be the same as that the variance of $\mathbf{x} - P\mathbf{x}$ is as small as possible. If we take $\mathbf{x}$ to be a column vector we can write the matrix P as $P = QQ^{\mathrm{T}}$ where the columns in Q are an orthonormal basis for the range of P so that $\mathbb{Q}^{\mathrm{T}}Q = I$. Thus the (euclidean) length of $P\mathbf{x}$ is equal to the length of $Q^{\mathrm{T}}\mathbf{x}$. The variance of $Q^{\mathrm{T}}\mathbf{x}$ can be written as $E(\mathbf{x}^{\mathrm{T}}QQ^{\mathrm{T}}\mathbf{x})$ or $E(\mathrm{trace}\,(Q^{\mathrm{T}}\mathbf{x}\mathbf{x}^{\mathrm{T}}Q))$, where E denotes the expectation (that is the integral or sum over the probability space). If we let $R = E(\mathbf{x}\mathbf{x}^{\mathrm{T}})$ we see that we should choose Q in such a way that $\mathrm{trace}\,(Q^{\mathrm{T}}RQ)$ is as large as possible. We have the following result.

Lemma 26. *Let R be a nonnegative definite symmetric $d \times d$ matrix with eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_d$. Then the maximum value of* $\mathrm{trace}\,(Q^{\mathrm{T}}RQ)$ *where Q is a $d \times m$ matrix with $Q^{\mathrm{T}}Q = I$ is $\sum_{j=1}^{m} \lambda_j$ and it is achieved when the columns in Q form an orthonormal basis of the space spanned by the eigenvectors associated with $\lambda_1, \ldots, \lambda_m$.*

Another way of looking at this result is to consider the case where one is given n vectors $\mathbf{x}_j$, $j = 1, \ldots, n$ such that $\sum_{j=1}^{n} \mathbf{x}_j = \mathbf{0}$. Then on can form a $d \times n$-matrix A with the vectors $\mathbf{x}_j$ as columns. If now we form the

singular value decomposition $A = USV^{\mathrm{T}}$. In this case $E(\mathbf{x}\mathbf{x}^{\mathrm{T}}) = \frac{1}{n}AA^{\mathrm{T}} = \frac{1}{n}USS^{\mathrm{T}}U^{\mathrm{T}}$ From this it follows that Q consists of the first m columns of U.

Next we consider the case where the mean value of $\mathbf{x}_j$ is not zero (or $E(\mathbf{x}) \neq \mathbf{0}$). In that case we calculate $\overline{\mathbf{x}} = \frac{1}{n}\sum_{j=1}^{n}\mathbf{x}_j$ and consider instead the vectors $\mathbf{x}_j - \overline{\mathbf{x}}$. If we formulate the problem in term of a minimization problem we have to calculate

$$\min_{Q}\sum_{j=1}^{n}\left|\mathbf{x}_j - Q(Q^{\mathrm{T}}\mathbf{x}_j - Q\overline{\mathbf{x}}) - \overline{\mathbf{x}}\right|^2.$$

It turns out that we get the same result by calculating

$$\min_{Q,\mathbf{y}_j,\mathbf{c}}\sum_{j=1}^{n}\left|\mathbf{x}_j - Q\mathbf{y}_j - \mathbf{c}\right|^2,$$

that is, we get $\mathbf{c} = \overline{\mathbf{x}}$ and $\mathbf{y}_j = Q^{\mathrm{T}}\mathbf{x}_j - Q\overline{\mathbf{x}}$. (Clearly $\mathbf{c}$ and $\mathbf{y}_j$ are not uniquely determined since one can always add a constant to all $\mathbf{y}_j$ and subtarsct it from $\mathbf{c}$.) One can even go further and formulate the following minimization problems:

$$\min_{f\in A(m,d),\mathbf{y}_j}\sum_{j=1}^{2}|\mathbf{x}_j - f(\mathbf{y}_j)|^2$$

$$\min_{g\in A(d,m),f\in A(d,m)}\sum_{j=1}^{2}|\mathbf{x}_j - f(g(\mathbf{x}_j))|^2,$$

where $A(k,n)$ is the set of all functions of the form $\mathbf{z}\in\mathbb{R}^k \mapsto A\mathbf{z}+\mathbf{c}\in\mathbb{R}^n$ where A is an $n\times m$ matrix.

2. Nonlinear principal components and neural networks

The theory of principal components briefly presented in the previous section does not really have anything to do neural networks, but every function in the set $A(k,n)$ can, of course be realized as a neural network with linear activation function. This gives immediately the possibility of defining several kinds of "nonlinear principal components" if one gives up the requirement that the funcions f and g are affine. Clearly, one cannot take arbitrary functions because then one could, using e.g. a space filling curve make the expression to be minimized equal to 0 in infinitely quite different ways so no information could be extracted. Neural networks, however, provide an easily parametrized class of functions that can be used for this purpose.

Here we just present an example of what one can get in a very simple case. In the picture below the dots are given data points. We use a neural

network with dimensions $[2, 5, 1, 5, 2]$ and the line drawn in the picture is the graph of the mapping from level 2 to level 4, that is points in the plane are "projected" onto the real line and reconstructed from the line with this mapping.