

Module 4

Reformulation of the objective function.

In the last lesson we determined the coordinates of the optimum projection w.r.t the orthonormal basis that spans our principal subspace

Before we go on to determine the optimum / mal basis vectors, let rephrase our loss function first.

The result that we have so far:

Ⓐ Description of our projected datapoint:

$$\hat{x}_n = \sum_{j=1}^M \beta_{jn} b_j$$

Ⓑ Our loss function:

$$J = \frac{1}{N} \sum_{n=1}^N \|x_n - \hat{x}_n\|^2$$

e

I 9369

③ The partial derivative of our loss function
w.r.t. our projected data point

2

$$\frac{\partial J}{\partial \hat{x}_n} = -\frac{2}{N} (\hat{x}_n - \tilde{x}_n)^T$$

① optimal coordinate that we found in last season.

$$\beta_{j,n} = \hat{x}_n b_j, j=1, \dots, m.$$

Before we go on, and determine the optimal basis vectors,

Let's rephrase our loss function.

We will make it much easier to find our basis vectors.

For this, let's have a closer look at the difference vector between our original data point and our projected data point.

"We can write X_n is given by equation A, which is the sum $j=1$ to M , B_j times b_j ".

$$\stackrel{A}{=} \sum_{j=1}^M B_j b_j$$

Now we now use the results of our optimal B_j parameters, we get "from $j=1$ to M , X_n transpose times b_j times b_j " use ①

$$\stackrel{①}{=} \sum_{j=1}^M (X_n^T b_j) b_j$$

Now we rewrite this in the following way:

$(X_n^T b_j)$ is just a scalar or dot product in this particular case, and dot products are symmetric, so we can swap the order, and we can also move the scalar to end \downarrow see above

So what we end up with:

$$= \sum_{j=1}^M b_j (b_j^T x_n) = \left(\sum_{j=1}^m b_j b_j^T \right) x_n$$

more x_n out of sum

Projection matrix

(1)

and if we look at this (1), it's a projection matrix,
this means that \hat{x}_n is the orthogonal projection of

x_n onto the subspace spanned by the
 M basis vectors b_j where $j = 1, \dots, M$

$$\left(\sum_{j=1}^m b_j b_j^T \right)$$

Similarly we can write $x_n = \sum_{j=1}^M b_j b_j^T x_n$ of

b_j times b_j transpose times x_n

$$x_n = \left(\sum_{j=1}^m b_j b_j^T \right) x_n \quad \leftarrow \textcircled{X}$$

$$+ \left(\sum_{j=M+1}^N b_j b_j^T \right) x_n \quad \leftarrow \textcircled{Y}$$

→ we write x_n as a projection onto the principal subspace plus ⑨ a projection onto the orthogonal complement

(Remember $\hat{x}_n \leftarrow$ the approximation onto x_n)

So we now look at the difference vector between \hat{x}_n and x_n , what remains is exactly the term ⑨ page 4

$$\therefore \underline{\underline{x_n - \hat{x}_n}} = \underbrace{\left(\sum_{j=m+1}^n b_j b_j^T \right) x_n}_{\text{⑩}}$$

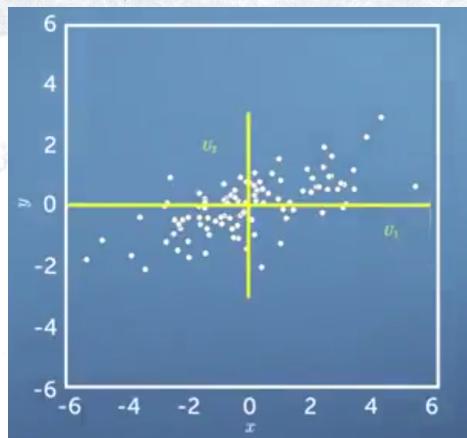
So now we can look at this displacement vector, the difference between x_n and its projection (\hat{x}_n),

and we can see that the displacement vector, is exclusively in the subspace we

ignore, i.e. ⑩

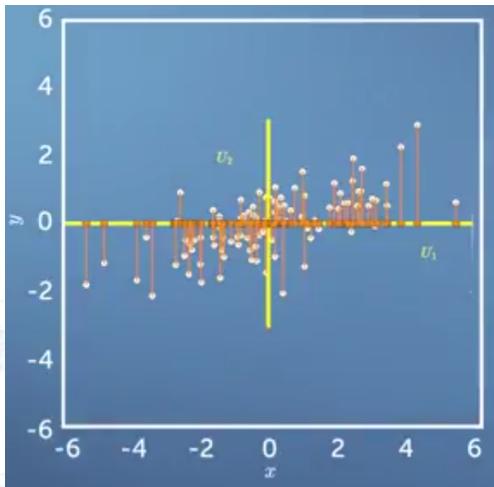
that means, the orthogonal complement to
the principal subspace.

Let's look at an example in 2 Dim. (see pg)



We have a dataset in two Dim, represented by
these dots, and now we interested in projecting
them onto the u_1 subspace.

When we do this and look at the difference vector,
between the original ~~vector~~ and ^{data} projected data,
we get these vertical lines, that means they
have no x component, no variation 'off x ',
that means they only have a component that
lives in subspace u_2 , which is the
orthogonal complement to u_1 , which
is the subspace that we projected onto (space)



so with this illustration, let's quickly rewrite this
in a slightly different way:

$$= \sum_{j=M+1}^n (b_j^T x_n) b_j \quad \text{equation } E$$

We looked at the displacement vector between x_n
and its orthogonal projection onto the principal
subspace X_n

and now we're going to use this to reformulate
our loss function

So from equation (B),
 "we get that our loss function is lower N
 times Sum n=1 to N, of $x_n - \hat{x}_n$ squared,
 average squared reconstruction error."

$$J = \frac{1}{N} \sum_{n=1}^N \|x_n - \hat{x}_n\|^2$$

And now we going to use equation (E) after
 the displacement vector we (F)

We rewrite the now, using equation (G):

"as 1 of N times Sum n=1 to N, and now
 we going have inside that squared norm, the
 we going have inside that squared norm, the
 expression we (E); sum j=m+1 to D of
 b_j transpose times x_n times b_j squared."

$$E = \frac{1}{N} \sum_{n=1}^N \left\| \sum_{j=m+1}^D (b_j^T x_n) b_j \right\|^2$$

And now we going to use the fact that the b_j form
an orthonormal basis which will greatly
simplify the expression:

$$\text{ONB} = \frac{1}{N} \sum_{n=1}^N \sum_{j=m+1}^D (b_j^T x_n)^2$$

identical.

And now we going to multiply it out explicitly

$$= \frac{1}{N} \sum_n \sum_j b_j^T x_n * x_n^T b_j$$

Now we rearrange the sum, we going to move
the sum over j outside:

$$= \sum_{j=m+1}^D b_j^T \left(\frac{1}{N} \sum_{n=1}^N x_n x_n^T \right) b_j$$

Now if we look very carefully we can identify
this expression  (above)

10
the data Covariance matrix S because
we assumed we have centered the data.

So mean of data is 0

This means now we can rewrite our loss
function using the data Covariance matrix
and we get: , our loss is the ... sum ...

$$= \sum_{j=M+1}^n b_j^T S b_j$$

we can also use a slightly different interpretation.
by rearranging a few terms using
the trace operator

$$= \text{trace} \left(\left(\sum_{j=M+1}^n b_j^T b_j \right) S \right)$$

projector

Can also interpret this matrix as a
projection matrix

(11)

The projection matrix, takes our data covariance matrix and project it onto the orthogonal complement of the principal subspace.

That means we can reformulate our loss function as the variance of the data projected onto the ~~the~~ subspace that we ignore

Therefore minimizing the loss, is equivalent to minimizing the variance of the data that lies in the subspace that's orthogonal to the principal subspace

In other words, we interested in retaining as much variance after projection as possible.

The reformulation of the average squared reconstruction error i.e. data covariance gives us an easy way to find a basis vector of the principal subspace, which we will do in next lesson.