

PCA: Correction

I think there should be slight corrections in experiment.py related to how we shift to mean and project vectors from pixel space to eigenspace.

Corrections:

1. Shift to mean:

'experiment.py' calculates μ (mean) for both training and testing. I think μ should be common between training and testing, as μ is a property of transformation. So it should just be calculated from training set and applied to test set directly. That is,

```
mu = ps6.get_mean_face(Xtrain)
...
# testing
mu = ps6.get_mean_face(Xtest)
Xtest_proj = np.dot(Xtest - mu, eig_vecs)
```

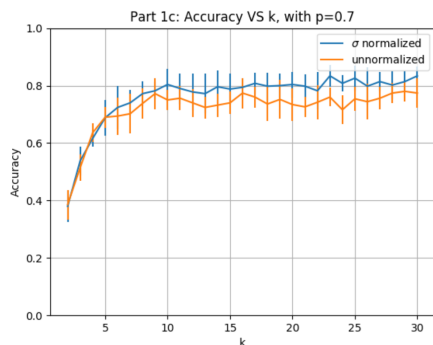
2. Normalize projected vector in eigenspace by standard deviations ($\sqrt{\text{eigenvalue}}$'s) in corresponding dimensions.

This might be tricky to understand, as lectures have also not pointed this out.

To motivate the claim that this should be done, please see my comparison plot.

Orange line is straight as per 'experiment.py' (with μ correction as explained above).

Blue is when I also normalize with σ 's (standard deviations) in corresponding dimensions.



TL;DR

we were earlier just doing (after taking top 'k' eigenvectors)

$A^T \cdot (X - \mu)$... i.e. simply projecting $(X - \mu)$ onto eigenvectors.

But we should do,

$\Lambda^{-1/2} \cdot A^T \cdot (X - \mu)$... i.e. projecting $(X - \mu)$ onto eigenvectors and normalizing by $\Lambda^{1/2}$

where,

A: Matrix of eigenvectors, such that $A^T \cdot A = I$

Λ : Diagonal Matrix of eigenvalues.

Proof:

Now why would this make sense? Intuitively, consider 2 probability distributions A and B and let's say both are Gaussians.

Case A:

$\mu=0, \sigma=10$

a point $P_A=20$ is 2 σ 's away from the mean of distribution A.

Case B:

$\mu=0, \sigma=100$

a point $P_B=100$ is 1 σ away from the mean of distribution B.

Question is: which point is closer to μ 's of respective distributions?

If you just consider Euclidean distance, you will say P_A . But that would be wrong as we are interested in how far a point is from the distribution. So in that sense, P_B is closer to the distribution.

Now, what does it have to do with PCA?

You might have noticed that PCA minimizes the Mean Squared Error (MSE) between original data (image in our case) and reconstructed data. It is well known that MSE minimization (like in linear regression) assumes a Gaussian prior on data.

Now since we are dealing with multiple dimensions, we need to consider multi-variate Gaussian distribution, which has following form.

$$N(X|\mu, \Sigma) = \frac{1}{\sqrt{\det(2\pi \cdot \Sigma)}} * e^{-\frac{(X-\mu)^T \Sigma^{-1} (X-\mu)}{2}}$$

Important consideration that comes out from multi-variate Gaussian distribution is Mahalanobis distance = $\sqrt{(X-\mu)^T \cdot \Sigma^{-1} \cdot (X-\mu)}$.

This is a dimensionless quantity and incorporates $(X - \mu)$ normalization by $\Sigma^{1/2}$ as I show below.

$$(X - \mu)^T \cdot \Sigma^{-1} \cdot (X - \mu) = (X - \mu)^T \cdot (\Sigma^{-1/2}) \cdot (\Sigma^{-1/2})^T \cdot (X - \mu)$$

(where, $\Sigma^{-1/2} = L$ is Cholesky decomposition of Σ^{-1})

$$= (L^T \cdot (X - \mu))^T \cdot (L^T \cdot (X - \mu))$$

Now, $L^T \cdot (X - \mu)$ is projection of $X - \mu$ onto L , which is similar to our familiar projection in eigenspace of Σ , but also normalized by $\sqrt{\text{eigenvalue}}$'s as I show below.

Since Σ is a symmetric matrix (and hopefully positive definite matrix),

$\Sigma = A \cdot \Lambda \cdot A^T$.. (can be done via SVD decomposition)

where,
A: Matrix of eigenvectors, such that $A^{-1} = A^T$
 Λ : Diagonal Matrix of eigenvalues.

Since Λ is diagonal matrix (all positive values as we assumed Σ is positive definite), we can simply take square root of its elements and write it as,
 $\Lambda = (\Lambda^{1/2}). (\Lambda^{1/2})^T$

Now,
 $\Sigma = A. \Lambda. A^T = (A. \Lambda^{1/2}). (A. \Lambda^{1/2})^T$
 $\rightarrow \Sigma^{-1} = ((\Lambda^{1/2})^{-1}. A^{-1})^T. ((\Lambda^{1/2})^{-1}. A^{-1})$
 $\rightarrow \Sigma^{-1} = (\Lambda^{-1/2}. A^T)^T. (\Lambda^{-1/2}. A^T) = L. L^T$
 $\rightarrow L^T = \Lambda^{-1/2}. A^T$... this basically mean multiply each eigenvector (row of A^T) with its $\frac{1}{\sqrt{eigenvalue}}$, exactly what we wanted.

Hence,
we were earlier just doing (after taking top 'k' eigenvectors)
 $A^T. (X - \mu)$... i.e. simply projecting $(X - \mu)$ onto eigenvectors.
But we should do,
 $\Lambda^{-1/2}. A^T. (X - \mu)$... i.e. projecting $(X - \mu)$ onto eigenvectors and normalizing by $\Lambda^{1/2}$

Q.E.D

ps6 module8 bugs


Updated 1 month ago by Yogesh Luthra

the students' answer, where students collectively construct a single answer

Click to start off the wiki answer

followup discussions for lingering questions and comments


☐ Resolved ☒ Unresolved


 **Yi Sun** 1 month ago
Hi Yogesh, thanks for writing this up. This was very thoughtful.

This is like calculating the normalized $Z_score = (X-u)/sigma$, which makes a lot of sense for comparison normalized distributions (rather than just euclidean distance from mean). I am inclined to believe your proposed method has better accuracy as you have shown in the chart.


Could instructor/TAs please help weigh in on this topic? It would be greatly appreciated.

Thanks,
Yi

 **Yogesh Luthra** 1 month ago
Yi, your intuition is exactly right, that z_score is being found here.

 **Yi Sun** 1 month ago
I also echo your thought on using the training data mean during testing. Using mean calculated from testing data is like using eigenvectors calculated from testing data during prediction, which would be quite awkward. If we are using the normalized z_score method, we definitely need to use the mean vector that correspond to the sigma vector, and both of these are calculated from the training data along with the eigenvectors.

☐ Resolved ☒ Unresolved

 **Michael Perry** 25 days ago
Here is a [video](#) that helped me understand the difference between Mahalanobis distance and Euclidean distance and why Mahalanobis distance is preferred in this case.