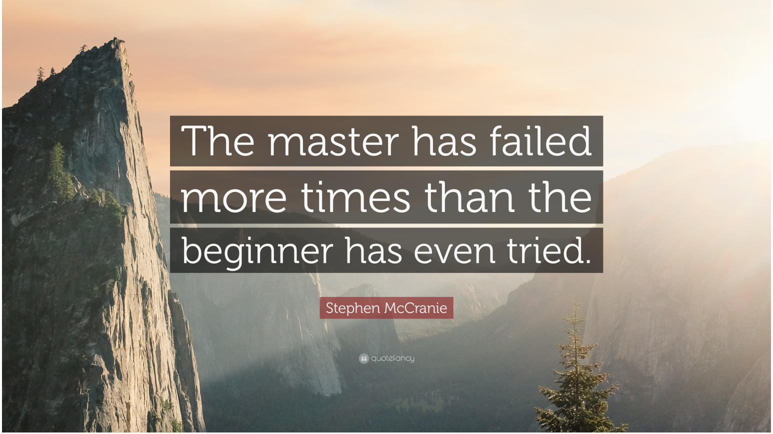
EECS 551 Discussion 10

Task 5 - Subspace Learning

Eric Cheek

November 12, 2021

Don't be discouraged, keep going!



Background: Task 2

Remember in Task 2, we wanted to solve the problem:

$$\hat{c} = \underset{c \in \{1,\ldots,C\}}{\arg\min}\; d_c(\boldsymbol{v}), \quad d_c(\boldsymbol{v}) = \left\|\boldsymbol{v} - \boldsymbol{U}_c(\boldsymbol{U}_c'\boldsymbol{v})\right\|_2 = \left\|\boldsymbol{P}_{\mathcal{S}_c}^{\perp}\boldsymbol{v}\right\|_2 .$$

$\hat{c}$ is the predicted class label for test image $\mathbf{v}$

$\mathbf{U}_c$ is an orthonormal basis for the subspace defined by training samples in class c

$d_c(\mathbf{v})$ is the distance from $\mathbf{v}$ to it's projection onto the subspace that is spanned by the columns of $\mathbf{U}_c$

Background - Task 2

Remember, we found $\mathbf{U}_c$ by taking an SVD of our training data.

For example, if $\mathbf{X}_1$ defined a matrix whose columns were training samples (they were images in our case), then

$$\mathbf{X}_1 = \mathbf{U\Sigma V}'$$

Depending on the number of relevant singular values (as determined by a scree plot, for example), we can heuristically find the number of left singular vectors to create an orthonormal basis $\mathbf{U}_1 \in \mathbb{F}^{MxK}$

We can repeat this process for all of our c classes.

Subspace Learning

Goal: Given several data matrices $\mathbf{X}_j$, learn a subspace for each of the j classes.

$$\underbrace{\boldsymbol{X}}_{M \times N} \approx \underbrace{\boldsymbol{Q}}_{M \times K} \underbrace{\boldsymbol{Z}}_{K \times N}, \quad \text{where } \boldsymbol{Z} \triangleq \begin{bmatrix} \boldsymbol{z}_1 & \dots & \boldsymbol{z}_N \end{bmatrix}.$$

To find $\boldsymbol{Q}$ and $\boldsymbol{Z}$ we could pursue the following optimization problem:

$$\hat{\boldsymbol{Q}} = \underset{\boldsymbol{Q} \in \mathbb{F}^{M \times K}}{\arg\min} \min_{\boldsymbol{Z} \in \mathbb{F}^{K \times N}} \|\boldsymbol{X} - \boldsymbol{Q}\boldsymbol{Z}\|_{\mathrm{F}}, \quad \text{s.t. } \boldsymbol{Q}'\boldsymbol{Q} = \boldsymbol{I}_K. \tag{6-15}$$

In this setting, typically $K \ll \min(M, N)$ so the product $\boldsymbol{Q}\boldsymbol{Z}$ is matrix with (at most) rank K.

Thus this problem is essentially a low-rank approximation problem except that here we really care only about the subspace basis $\boldsymbol{Q}$ and not the coefficients $\boldsymbol{Z}$ (nor their product). The low-rank solution is

$$\hat{\boldsymbol{X}}_K = \sum_{k=1}^{K} \sigma_k \boldsymbol{u}_k \boldsymbol{v}_k' = \boldsymbol{U}_K \boldsymbol{\Sigma}_K \boldsymbol{V}_K' = \underbrace{\boldsymbol{U}_K}_{\hat{\boldsymbol{Q}}} \underbrace{\boldsymbol{\Sigma}_K \boldsymbol{V}_K'}_{\boldsymbol{Z}}.$$

Task 5 Part 1 - Data Reshaping

When we load our data:

- Each image is size nx*ny $= 28 * 28 = 784$ pixels total
- ntrain $= 50$, ntest $= 60$, ndigits $= 3$.
- size(testl) = $784, 60, 3$

Use the above information to create train by reshaping testl.

Use the above to define n,p,m,d in part 1 A. Do not assign them numerically, but rather define them in terms of size(test) and size(train).

Task 5 Part 1 - Constructing Basis for each digit

Our goal now is to define $\mathbf{U_c}$ for each of our c digits. K is defined earlier in the code to be 5.

We want to extract the first K left singular vectors of our training data corresponding to each digit d.

size(train) $=$ nx*ny, ntrain, d. You will need to compute an SVD across each of these d matrices.

Task 5 Part 1 - Computing distance vectors

Now that we have our d orthonormal basis for each of the d subspaces, we want to see how far each test image is from each subspace.

$$d_c(\mathbf{v}) = \|\mathbf{v} - \mathbf{U_c}(\mathbf{U_c'}\mathbf{v})\|_2^2$$

Let $\mathbf{v_i}$ denote the ith column of test data $\mathbf{V}$.

$$\mathbf{V} - \mathbf{U_c}(\mathbf{U_c'}\mathbf{V}) = [\mathbf{v}_1, ..., \mathbf{v_n}] - \left[\mathbf{U_c}(\mathbf{U_c'}\mathbf{v}_1), ..., \mathbf{U_c}(\mathbf{U_c'}\mathbf{v_n})\right]$$

From here, taking the elementwise sum of this result and then summing across the columns gives the distance between each training sample and the dth subspace.

Task 5 Part 1 - Assigning labels based on this distance

For classification, we say a test sample belongs to the class whose subspace is the closest.

If $d_1(\mathbf{v}), d_2(\mathbf{v}), ..., d_d(\mathbf{v})$ denote the distances of a sample to the different subspaces, then:

$$\hat{c} = \underset{c \in \{1,...,d\}}{\operatorname{argmin}} \ d_c(\mathbf{v})$$

Task 5 Part 2 - Put it all together

In part 2, we essentially take all of the code we wrote and copy and paste it into a function called classify image.

This is why we defined variables n, p, m, d earlier. We wanted our code to be generic so it is easily generalizable to data of other sizes!

Task 5 Part 3 - Try multiple values of K

In part 3, we run our function for classification on all digits, not just 3.

We reshape testr the same way as we did with test in part A.

Task 5 Part 4 - Visually determine a reasonable value of K

In part 4, we will determine a reasonable value of K by looking at the scree plots of singular values for at least two different sets of images.

Practice Questions!

- Is the orthonormal basis for a subspace unique?
- Why is classification here more efficient than using kNN?