The Mahalanobis Distance in Character Recognition

Authored by Dan Frey 31 July 98

This sheet explores the use of the Mahalanobis Distance in character recognition.

It defines a letters A, B, C, and D.
It creates a population of distorted letters to train a classifier.
It creates a test population of letters to test the classifier.
It allows you to compete against the Mahalanobis classifier.

$\mathrm{ORIGIN} := 1$ Every matrix and vector in this sheet will begin with index number 1.

Here are the Canonical letters A B C and D defined as matrices of 1s and 0s

$$A := \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix} \quad B := \begin{pmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix} \quad C := \begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix} \quad D := \begin{pmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix}$$

Stretch the letters out into a vector of "features".

$k := 1..25$

$$\mathrm{Alin}_k := A_{k+5-5\cdot \mathrm{ceil}\left(\frac{k}{5}\right), \mathrm{ceil}\left(\frac{k}{5}\right)} \qquad \mathrm{Blin}_k := B_{k+5-5\cdot \mathrm{ceil}\left(\frac{k}{5}\right), \mathrm{ceil}\left(\frac{k}{5}\right)}$$

$$\mathrm{Clin}_k := C_{k+5-5\cdot \mathrm{ceil}\left(\frac{k}{5}\right), \mathrm{ceil}\left(\frac{k}{5}\right)} \qquad \mathrm{Dlin}_k := D_{k+5-5\cdot \mathrm{ceil}\left(\frac{k}{5}\right), \mathrm{ceil}\left(\frac{k}{5}\right)}$$

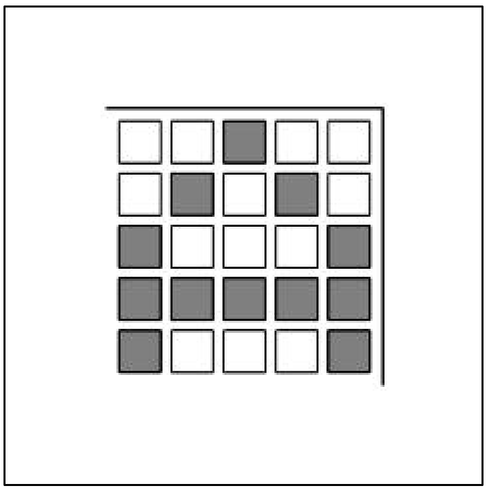
Reverse the letters. This makes the plots easier to read.

$\mathrm{Alin}_k := \mathrm{Alin}_k = 0 \quad \mathrm{Blin}_k := \mathrm{Blin}_k = 0 \quad \mathrm{Clin}_k := \mathrm{Clin}_k = 0 \quad \mathrm{Dlin}_k := \mathrm{Dlin}_k = 0$

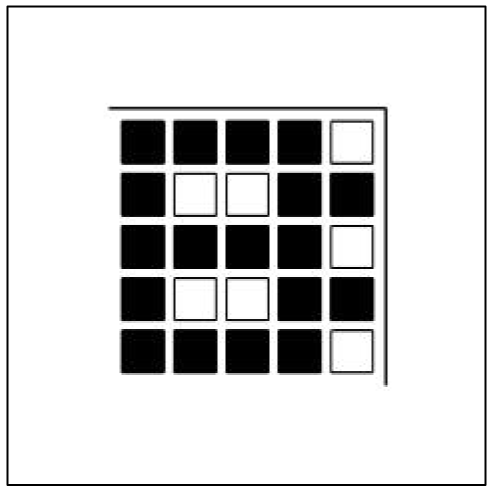
Define a function that will display the vector of features as a matrix

$$\mathrm{DISPLAY}(Llin) := \left|\begin{array}{l} \text{for } i \in 1..5 \\ \quad \text{for } j \in 1..5 \\ \qquad L_{i,j} \leftarrow Llin_{i+5\cdot(j-1)} \\ L \end{array}\right.$$

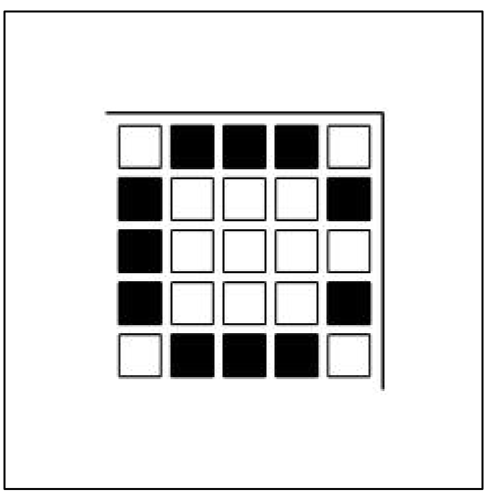
Here are a couple of plots that show what the Cononical letters look like.



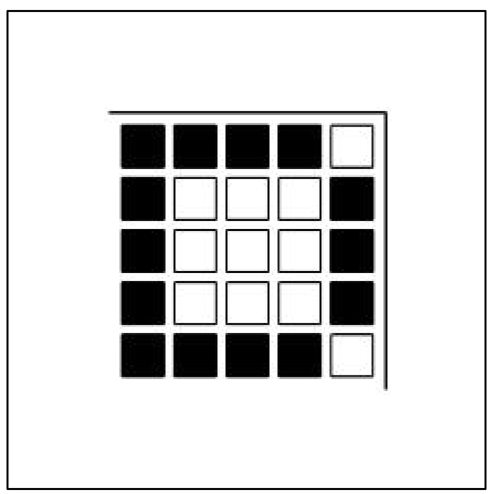
DISPLAY(Alin)



DISPLAY(Blin)



DISPLAY(Clin)



DISPLAY(Dlin)

Create a population of fuzzed up letters

$\text{training\_population} := 300$ Size of the training population. Has to be bigger than the number of pixels.

$\text{pop} := 1..\,\text{training\_population}$

$\sigma_t := 0.01$ How much random noise is superposed onto the pixels.

$$APOP^{\langle pop \rangle} := \overrightarrow{\left[\left(2\cdot \text{floor}\left(\text{runif}(1,0,2)_1\right) - 1\right)\cdot \text{Alin}\right]} + \text{rnorm}\left(25,0,\sigma_t\right)$$

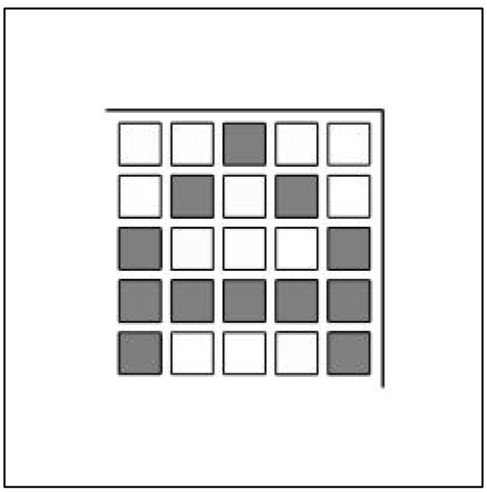
$$BPOP^{\langle pop \rangle} := \overrightarrow{\left[\left(2\cdot \text{floor}\left(\text{runif}(1,0,2)_1\right) - 1\right)\cdot \text{Blin}\right]} + \text{rnorm}\left(25,0,\sigma_t\right)$$

$$CPOP^{\langle pop \rangle} := \overrightarrow{\left[\left(2\cdot \text{floor}\left(\text{runif}(1,0,2)_1\right) - 1\right)\cdot \text{Clin}\right]} + \text{rnorm}\left(25,0,\sigma_t\right)$$

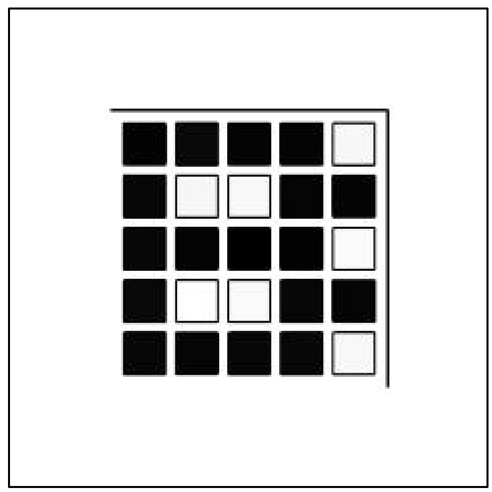
$$DPOP^{\langle pop \rangle} := \overrightarrow{\left[\left(2\cdot \text{floor}\left(\text{runif}(1,0,2)_1\right) - 1\right)\cdot \text{Dlin}\right]} + \text{rnorm}\left(25,0,\sigma_t\right)$$

These populations have been switched from positive to negative with a 50/50 probability, then superposed with white noise.

Here is an example of a fuzzed up A and a fuzzed up B.



$\text{DISPLAY}\left(APOP^{\langle 5 \rangle}\right)$



$\text{DISPLAY}\left(BPOP^{\langle 3 \rangle}\right)$

Compute Statistics of the Training Population

To create a Mahananobis distanace classifier, we need to deternime the mean and cavariance matrices from the population of letters.

$n := 1..25 \qquad m := 1..25$

$$COVA_{n,m} := cvar\left[\left(APOP^T\right)^{\langle n\rangle}, \left(APOP^T\right)^{\langle m\rangle}\right] \qquad MEANA_n := mean\left[\left(APOP^T\right)^{\langle n\rangle}\right]$$

$$COVB_{n,m} := cvar\left[\left(BPOP^T\right)^{\langle n\rangle}, \left(BPOP^T\right)^{\langle m\rangle}\right] \qquad MEANB_n := mean\left[\left(BPOP^T\right)^{\langle n\rangle}\right]$$

$$COVC_{n,m} := cvar\left[\left(CPOP^T\right)^{\langle n\rangle}, \left(CPOP^T\right)^{\langle m\rangle}\right] \qquad MEANC_n := mean\left[\left(CPOP^T\right)^{\langle n\rangle}\right]$$

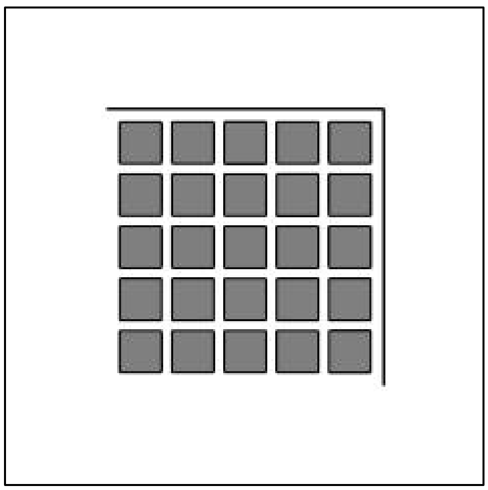
$$COVD_{n,m} := cvar\left[\left(DPOP^T\right)^{\langle n\rangle}, \left(DPOP^T\right)^{\langle m\rangle}\right] \qquad MEAND_n := mean\left[\left(DPOP^T\right)^{\langle n\rangle}\right]$$

The classifier will use the inverse of the covariance matrix, so let's compute it ahead of time.

$$INVCOVA := COVA^{-1} \qquad INVCOVB := COVB^{-1}$$

$$INVCOVC := COVC^{-1} \qquad INVCOVD := COVD^{-1}$$

I think it's interesting to note that the "mean" appearance of the letter A is incomprehensible as an A. The reason is that we had an equal number of positive images and negative images of A. The same applies to all the other letters. And yet the classifier successfully identifies the letter A because it has a map of the correlations within the letter A.



DISPLAY(MEANA)

Define the Classifier

Here is the classifier function. It is very simple. You send it a letter (L) as an argument. It computes the Mahalanobis distance from the letter to the class A, B, C, and D. Finally, it classifies L as the letter with the smallest Mahalanobis distance.

$$
\mathrm{Class}(L) := \left|
\begin{array}{l}
MDA \leftarrow (L - MEANA)^T \cdot INVCOVA \cdot (L - MEANA) \\
MDB \leftarrow (L - MEANB)^T \cdot INVCOVB \cdot (L - MEANB) \\
MDC \leftarrow (L - MEANC)^T \cdot INVCOVC \cdot (L - MEANC) \\
MDD \leftarrow (L - MEAND)^T \cdot INVCOVD \cdot (L - MEAND) \\
\text{"A"} \quad \text{if } (MDA < MDB) \cdot (MDA < MDC) \cdot (MDA < MDD) \\
\text{otherwise} \\
\quad \left|
\begin{array}{l}
\text{"B"} \quad \text{if } (MDB < MDC) \cdot (MDB < MDD) \\
\text{otherwise} \\
\quad \left|
\begin{array}{l}
\text{"C"} \quad \text{if } (MDC < MDD) \\
\text{"D"} \quad \text{otherwise}
\end{array}
\right.
\end{array}
\right.
\end{array}
\right.
$$

Test the Classifier

$\sigma := 0.6$ How badly should the letters be fuzzed up.

$tests := 10000$ How many times should you test the classifier

$test := 1..\,tests$

$$Answer_{test} := \left|\begin{array}{l} rand \leftarrow runif(1,0,4)_1 \\ \text{"A"} \quad \text{if } 0 \le rand \le 1 \\ \text{"B"} \quad \text{if } 1 < rand \le 2 \\ \text{"C"} \quad \text{if } 2 < rand \le 3 \\ \text{"D"} \quad \text{if } 3 < rand \le 4 \end{array}\right.$$

Make the four letters equally likely to appear within the test batch.

Create a batch of letters corresponding to the desired correct answers and appropriately fuzzed up. This will be the batch of data to test our classifier.

$$L_{test} := \left|\begin{array}{l} \left[\overrightarrow{\left(2\cdot floor\left(runif(1,0,2)_1\right) - 1\right)\cdot Alin}\right] + rnorm(25,0,\sigma) \quad \text{if } Answer_{test} = \text{"A"} \\ \left[\overrightarrow{\left(2\cdot floor\left(runif(1,0,2)_1\right) - 1\right)\cdot Blin}\right] + rnorm(25,0,\sigma) \quad \text{if } Answer_{test} = \text{"B"} \\ \left[\overrightarrow{\left(2\cdot floor\left(runif(1,0,2)_1\right) - 1\right)\cdot Clin}\right] + rnorm(25,0,\sigma) \quad \text{if } Answer_{test} = \text{"C"} \\ \left[\overrightarrow{\left(2\cdot floor\left(runif(1,0,2)_1\right) - 1\right)\cdot Dlin}\right] + rnorm(25,0,\sigma) \quad \text{if } Answer_{test} = \text{"D"} \end{array}\right.$$

$RIGHT_{test} := \left(Class\left(L_{test}\right) = Answer_{test}\right)$

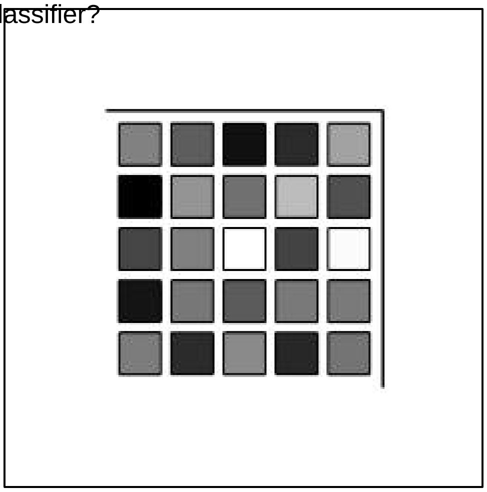
$mean(RIGHT) = 0.905$

At sigma=0.6, the Mahalanobis classifier is right about 94% of the time!
Can you do as well in classifying the letters?
Also note how fast the Mahalanobis classifier operates.

Try Your Hand at Classification

$Answ := 1$

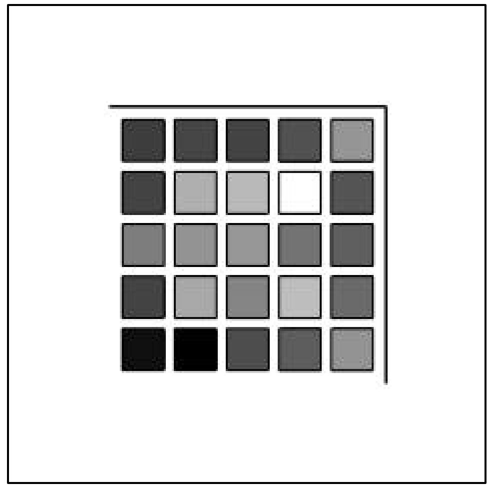
Take a guess at which letter is represented in the following four graphs. When you're done, change the varaible "Answ" to 1 and the answers will display. Did you get it right? Did the MD classifier?



$\mathrm{DISPLAY}(L_1)$

$Answer_{1\cdot(Answ=1)} = \text{"C"}$

$Class\left[L_{1\cdot(Answ=1)}\right] = \text{"C"}$

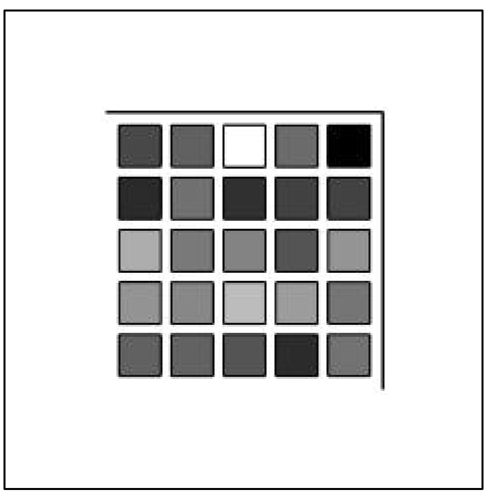


$\mathrm{DISPLAY}(L_2)$

D

$Answer_{2\cdot(Answ=1)} = \text{"D"}$

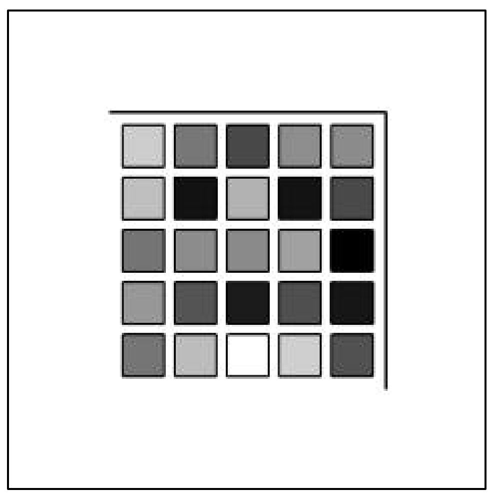
$Class\left[L_{2\cdot(Answ=1)}\right] = \text{"D"}$



$\mathrm{DISPLAY}(L_3)$

$Answer_{3\cdot(Answ=1)} = \text{"A"}$ A

$Class\left[L_{3\cdot(Answ=1)}\right] = \text{"A"}$



$\mathrm{DISPLAY}(L_4)$

A

$Answer_{4\cdot(Answ=1)} = \text{"A"}$

$Class\left[L_{4\cdot(Answ=1)}\right] = \text{"A"}$