

# Practice 11: Convolutions, Chi square and covariance.

Mayra Cristina Berrones Reyes 6291

November 17, 2020

## 1 Introduction

In this work we are going to discuss in three sections the subjects of convolution, chi squared and covariance with different examples. But first, we want to give a brief introduction into these three items.

### 1.1 Convolutions

Convolution refers to a mathematical operation on two functions ( $f$  and  $g$ ) that produce a third function ( $f * g$ ) that expresses how the shape of one is modified by the other. The term is often used both as the result function we mentioned, as well as the process of computing the function itself.

Convolutions can be find in several applications such as probability, statistics, computer vision, natural language processing, image and signal processing, engineering, etc.

The common engineering notational convention for this formula is shown in Equation 1,

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t - \tau)d\tau \quad (1)$$

■

### 1.2 Chi square test

The Pearson chi squared test is a statistical test applied to show the relationship between two categorical variables. This statistic is a single number that tells how much difference exists between the observed counts and the expected counts if there was no relationship at all in the population.

There are two types of chi squared tests:

- A chi squared goodness fit test, that determines if a sample data matches a population.
- A chi squared test to test for independence, comparing two variables in a contingency table to see if they are related, testing whether distributions differ from each other.

The formula for the chi squared statistic test used in this type of tests is shown in Equation 2,

$$x_c^2 = \sum \frac{(O_i - E_i)^2}{E_i} \quad (2)$$

■

Table 1: Some popularly known kernels for image processing.

Boxblur	Gaussian blur	Identity	Sharpen
$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$

### 1.3 Covariance

In mathematics and statistics, covariance is a measure of the relationship that two random variables have with each other. It evaluates how much the variables change together, but they do not measure the dependency between variables. The variance can take any positive or negative value. This values can be interpreted as:

- **Positive:** Indicates that two variables tend to move in the same direction.
- **Negative:** Reveals that two variables tend to move in an inverse direction.

The formula for the covariance between two random variables  $X$  and  $Y$  can be calculated using Equation 3,

$$Cov(X, Y) = \frac{\sum(X_i - \bar{X})(Y_j - \bar{Y})}{n} \quad (3)$$

■

## 2 Section 1: Convolutions

*Identify an application of your interest from the subject of convolution and present some related contribution (theoretical or numerical)*

Convolutions, as we see in the introduction of this work, are an extremely general idea. The expression used in Equation 1 can also be used in a higher number of dimensions. Just like with one dimensional convolutions, the concept of two dimensional convolutions can be imagined as sliding one function on top of another, multiplying and adding [7].

Our subject of interest, and one of the common application of convolutions is image processing. In this case, we think of the images as two dimensional functions, in the form of value of pixels. Many important image transformations such as noise additions, sharpening of the image, etc. are convolutions where we convolve the image function with a very small local function called “kernel”. The kernel slides to every position of the image and computes a new pixel as a weighted sum of pixels as it floats over.

In Table 1 we see some examples of popularly kernels used for different types of image processing. Depending on the scale it wants to work, we can change the size of its matrix and some of the numbers inside the matrix.

The easiest way of representing how exactly a kernel works, can be appreciated in Figure 1 where we can see the kernel sweeping over the image matrix. We multiply a batch of the image

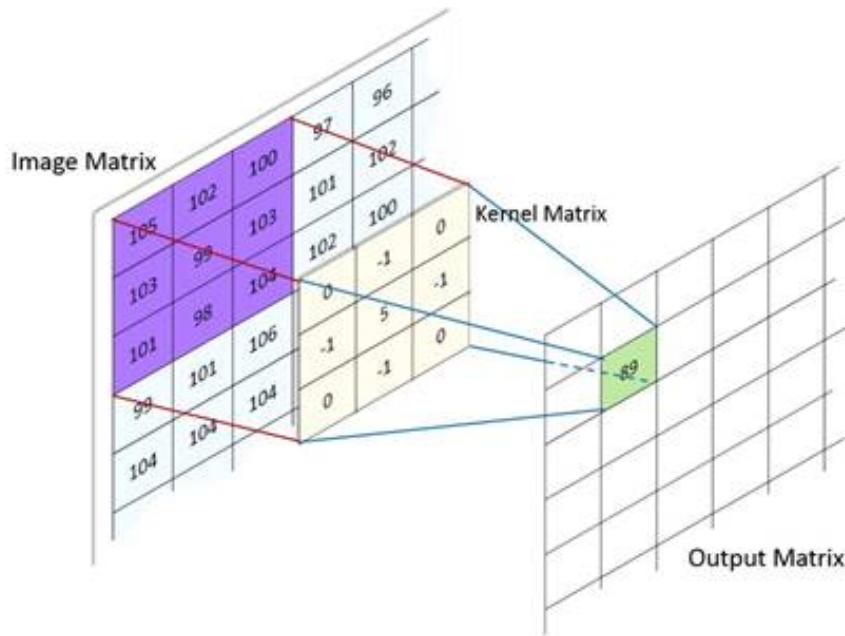


Figure 1: Graphical example of how a kernel works in a convolution of an image [6].

with the kernel, and then calculate the cumulative sum to add it to another matrix.

As an example of the use of convolutions in image processing, we can use the library of `magick` [4] in R that already has some integrated values for the kernels to add noise, edge detection, sharpen image, etc. In Figure 2 we see some of this examples. In this figure we also add a hand made kernel to blur the image, based on the values we used in Table 1.

### 3 Section 2: Chi square

*Identify an application of your interest (preferably of your thesis topic, if you have one) for the chi square test and apply it.*

For this application, following the subject of convolution, one of the main topics set for our thesis is the use of Convolutional Neural Networks (CNNs). For this type of Neural Network (NN) we use the same concept that we explained in the convolution section. We take the formula shown in Equation 4 and use it to calculate the index and rows of our resulting matrix.

$$G[m, n] = (f * h)[m, n] = \sum_j \sum_k h[j, k] f[m - j, n - k] \quad (4)$$

In the case of Convolutional Neural Networks (CNNs) we form a subsequent map of features in each convolution. The kernel used starts with random variables that get fitted in every iteration,

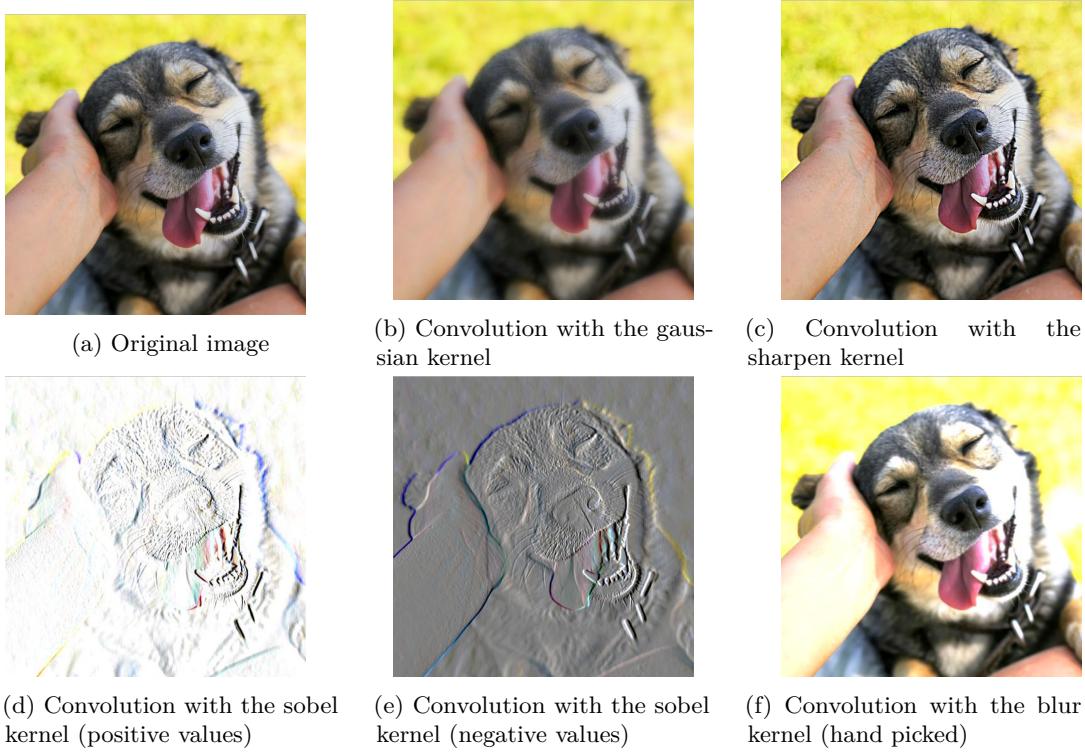


Figure 2: Use of convolution to change the an image of a dog.



Figure 3: Some images form the data set Mnist Fashion.

depending on the optimizer that we are using.

As we can see in Figure 1 if we, for example, started with a matrix of  $6 \times 6$  with a kernel of  $3 \times 3$ , we will get a feature map of  $4 \times 4$ . Since our image shrinks every time we perform a different convolution, we can only perform them a limited number of times before our image disappears completely. That is why there are many architectures of CNN and we experiment to find the best fit for our data.

For the experimentation part of this section, first we have to produce data that can be evaluated by the chi squared test. Since the images and algorithm we are currently working on is a little bit too heavy for the computer we are currently working on, we are using a famous (and digestible for regular computers) data set of images called MNIST Fashion [5], which is widely used as a benchmark in the data science community to validate their algorithms [3].

We run a simple CNN with the help of the library of **keras**. In Figure 2 we show some of the images contained in this data set, and in Table 2 we have the results of the model testing, and in Table 3 we show the confusion matrix for these classification.

To be able to use the confusion matrix represented in Table 3 we accommodate the data in a more compact table, as the one shown in Table 4, where we put the right and wrong predictions of each class in two different rows.

Table 2: Metric results of the CNN model for the classes from the MNIST Fashion data set.

Class	0	1	2	3	4	5	6	7	8	9
Sensitivity	88.5%	98.3%	82.4%	90.8%	75.5%	98.3%	67.5%	93.9%	94.4%	93.7%
Specificity	97.4%	99.7%	97.5%	98.9%	98.4%	99.2%	97.0%	99.3%	99.9%	99.7%
Pos Pred Value	79.0%	97.0%	78.6%	90.4%	84.1%	93.7%	71.3%	93.8%	98.6%	97.4%
Neg Pred Value	98.7%	99.8%	98.0%	99.0%	97.3%	99.8%	96.4%	99.3%	99.4%	99.3%
Prevalence	9.9%	10.1%	10.0%	9.9%	10.1%	10.1%	10.0%	9.9%	9.8%	10.2%
Detection Rate	8.8%	9.9%	8.3%	9.0%	7.6%	10.0%	6.7%	9.3%	9.3%	9.5%
Detection Prevalence	11.1%	10.2%	10.5%	10.0%	9.0%	10.6%	9.5%	9.9%	9.4%	9.8%
Balanced Accuracy	92.9%	99.0%	89.9%	94.9%	86.9%	98.8%	82.3%	96.6%	97.1%	96.7%

Table 3: Confusion matrix of prediction results for the CNN model. The principal diagonal represent the correct predictions, and all the other are the wrong predictions.

	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>
<b>0</b>	1052	1	26	29	1	0	209	0	13	1
<b>1</b>	5	1188	3	18	5	0	5	0	1	0
<b>2</b>	15	0	990	2	156	0	90	0	7	0
<b>3</b>	18	17	11	1083	37	0	23	0	9	0
<b>4</b>	3	1	67	33	910	0	54	0	14	0
<b>5</b>	1	0	0	0	0	1195	0	50	11	19
<b>6</b>	91	1	102	28	97	0	809	0	6	0
<b>7</b>	0	0	0	0	0	13	0	1113	4	57
<b>8</b>	4	0	3	0	0	0	8	1	1115	0
<b>9</b>	0	0	0	0	0	8	0	21	1	1141

Table 4: Table simplifying the data of the confusion matrix made in R.

Class	( , 0)	( , 1)	( , 2)	( , 3)	( , 4)	( , 5)	( , 6)	( , 7)	( , 8)	( , 9)
True predictions	1052	1188	990	1083	910	1195	809	1113	1115	1141
False predictions	277	37	270	115	172	81	325	72	16	30

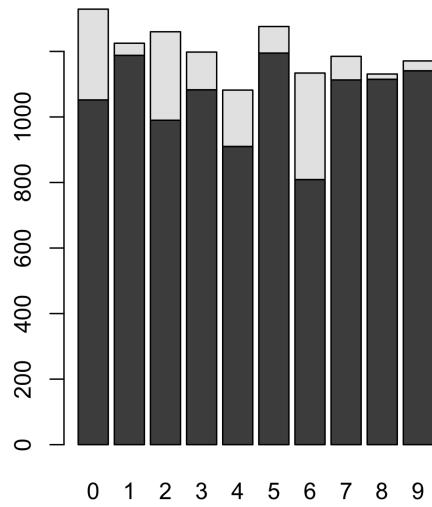


Figure 4: Barplot of the correct and incorrect predictions of the CNN model on each class.

Using the `chisq.test [1]` function in R we get as a result:

- **X squared:** 938.25
- **df:** 9
- **p value:**  $< 2.2 \times 10^{-16}$

So as a conclusion we can say that, in our example, the row and the column variables are statistically significantly associated. And finally, Figure 4 shows a bar plot of how the different classes are distributed, being the darker color the correct predictions, and the grey the wrong predictions.

## 4 Section 3: Covariance

*Validate the two properties related to the covariance that comes in the class page (preferably, first establish numerically that they are true and then prove them analytically)*

Both Equation 5 and 6 come from the example in the class page indicated in the instructions. For Equation 6 first we attempt a practical experiment in R using the functions of variance and covariance already in the program, as well as the function of `sample` to generate the random variables.

$$\text{Cov}[aX + b, cY + d] = ac\text{Cov}[X, Y] \quad (5)$$

$$\text{Var}[X + Y] = \text{Var}[X] - \text{Var}[Y] + 2\text{Cov}[X, Y] \quad (6)$$

In Figure 5 we have the different number of iterations. We went from 10, to 100 to 1000 iterations, in each one we tried the equation with a different number of random variables inside the variance and covariance functions, which can be seen in the x axis of all the plots.

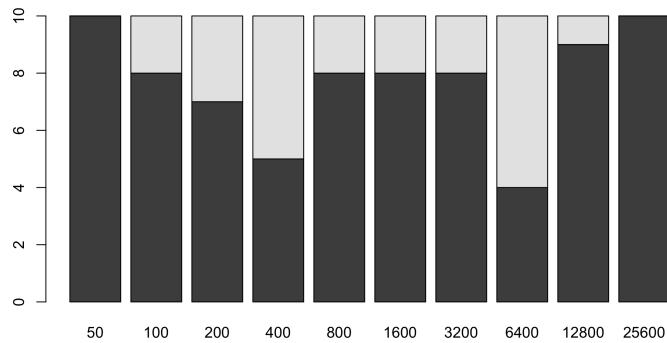
The darker color is for all the times the condition of equal was accomplished, and the gray color is when this condition was not met.

The analytic view of the formula can be described by Equation 7 [2],

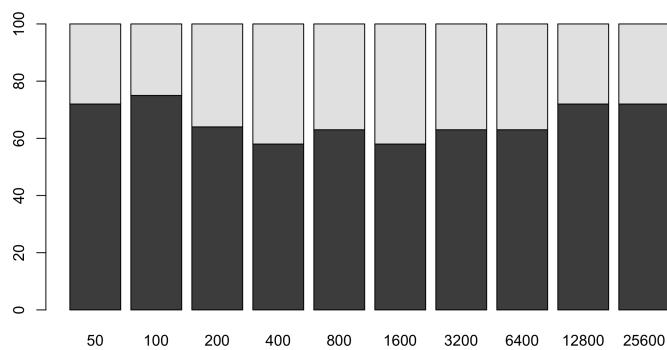
$$\begin{aligned} \text{Var}[X + Y] &= \sum_x \sum_y (x + y)^2 P_{XY}(x, y) - (E(X + Y))^2 \\ &= \sum_x \sum_y x^2 P_{XY}(x, y) + \sum_x \sum_y 2xy P_{XY}(x, y) + \\ &\quad \sum_x \sum_y y^2 P_{XY}(x, y) - (E(X))^2 - 2E(X)E(Y) - (E(Y))^2 \\ &= \sum_x x^2 P_X(x) - (E(X))^2 + \sum_y y^2 P_Y(y) - (E(Y))^2 + \\ &\quad \sum_x \sum_y 2xy P_{XY}(x, y) - 2E(X)E(Y) \\ &= E(X^2) - (E(X))^2 + E(Y^2) - (E(Y))^2 + 2(E(XY) - E(X)E(Y)) \\ &= \text{Var}(X) + \text{Var}(Y) + 2\text{Cov}(X, Y). \end{aligned} \quad (7)$$

## 5 Conclusion

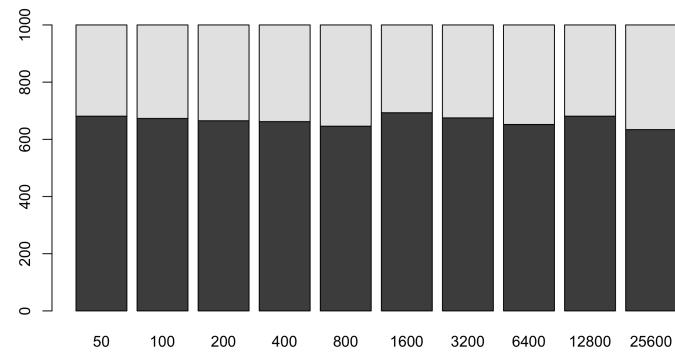
As always, I understand better all of this concepts when they come in the form of a practical example. In the reading part leading up to the class I was really confused by the concepts, and the explanations in class were not enough to dispel my doubts. It wasn't until some videos and examples later that I landed on the application of image processing. I thought this convolution and the one seen in class where two different concepts (one for statistics and the other for computer



(a) Barplot from 10 iterations of the equation comparision



(b) Barplot from 100 iterations of the equation comparision



(c) Barplot from 1000 iterations of the equation comparision

Figure 5: Barplots from the different iterations of the experiment of covariance.

science) but after seeing the application on the example, it became so easy to understand the equations seen class and its uses.

In the section of the Chi squared we used the MNIST dataset because the computer used for the experiments is of average processing capacity, and the images used in the real thesis problem are too big for it to be able to compile the program.

## References

- [1] Chi square test of independence in r. <https://www.statsandr.com/blog/chisquare-test-of-independence-in-r/>. Accessed: 2020-11-17.
- [2] Sums of random variables. <http://www.milefoot.com/math/stat/rv-sums.htm>. Accessed: 2020-11-17.
- [3] Lbb neural network. <https://www.kaggle.com/stvenl/lbb-neural-network>. Accessed: 2020-11-17.
- [4] Image convolution in r using magick. <https://ropensci.org/technotes/2017/11/02/image-convolve/>. Accessed: 2020-11-17.
- [5] Fashion mnist. <https://www.kaggle.com/zalando-research/fashionmnist>. Accessed: 2020-11-17.
- [6] Understanding convolutions. <https://colah.github.io/posts/2014-07-Understanding-Convolutions/>. Accessed: 2020-11-17.
- [7] Gentle dive into math behind convolutional neural networks. <https://towardsdatascience.com/gentle-dive-into-math-behind-convolutional-neural-networks-79a07dd44cf9>. Accessed: 2020-11-17.