Measures of similarity and summary statistics with MATLAB

Objective: The overall objective is to get a basic understanding for measures of similarity as well as summary statistics. Upon completing this exercise it is expected that you:

- Understand how to calculate summary statistics such as mean, variance, median, range, covariance and correlation.
- Understand the various measures of similarity such as Jaccard and Cosine similarity and apply similarity measures to query for similar observations.
- Understand the univariate and multivariate normal distribution.

Material: Pang-Ning Tan, Michael Steinbach, and Vipin Kumar, *"Introduction to Data Mining"* 2.4 + 3.1-3.2 + C1-C2 as well as the files in the exercise 3 folder available from Campusnet.

Preparation: Exercises 1-2 **MATLAB Help:** MATLAB help is obtained by typing `helpdesk` at the MATLAB command prompt or by typing `help <function name>` in the command prompt. In practice, the fastest and easiest way to get help in Matlab is often to simply Google your problem. For instance: `"How to add legends to a plot in Matlab"` or the content of an error message. In the later case, it is often helpful to find the *simplest* script or input to script which will raise the error.

Software installation: Extract the Matlab toolbox from Campusnet. Start Matlab and go to the `<base-dir>/02450Toolbox_Matlab/` directory using the command `cd('<base-dir>/02450Toolbox_Matlab/')` and run `setup.m`. Consider storing your scripts for exercise 3 in: `<base-dir>/02450Toolbox_Matlab/week3/`. **Representation of data in Matlab:**

	Matlab var.	Type	Size	Description
	`X`	Numeric	$N \times M$	Data matrix: The rows correspond to N data objects, each of which contains M attributes.
	`attributeNames`	Cell array	$M \times 1$	Attribute names: Name (string) for each of the M attributes.
	`N`	Numeric	Scalar	Number of data objects.
	`M`	Numeric	Scalar	Number of attributes.
Classification	`y`	Numeric	$N \times 1$	Class index: For each data object, `y` contains a class index, $\mathtt{y}_n \in \{0, 1, \ldots, C-1\}$, where C is the total number of classes.
Classification	`classNames`	Cell array	$C \times 1$	Class names: Name (string) for each of the C classes.
Classification	`C`	Numeric	Scalar	Number of classes.

3.1 Summary Statistics

3.1.1 Calculate the (empirical) mean, variance, median and range of the following set of numbers:

$$\{-0.68, -2.11, 2.39, 0.26, 1.46, 1.33, 1.03, -0.41, -0.33, 0.47\}$$

Hints:

- *Take a look at the functions* `mean`, `var`, `median`, *and* `range`

Run the script `ex3_1_1.m` to verify your calculations.

3.2 Measures of similarity

We will use a subset of the data on wild faces described in [1] transformed to a total of 1000 gray scale images of size 40 × 40 pixels, we will attempt to find faces in the data base that are the most similar to a given query face. To measure similarity we will consider the following measures: SMC, Jaccard, Cosine, ExtendedJaccard, and Correlation. These measures of similarity are described in ”Introduction to Data Mining” page 73-77 and are given by

$$\begin{aligned}
\text{SMC}(\boldsymbol{x},\boldsymbol{y}) &= \frac{\text{Number of matching attribute values}}{\text{Number of attributes}}\\
\text{Jaccard}(\boldsymbol{x},\boldsymbol{y}) &= \frac{\text{Number of matching presences}}{\text{Number of attributes not involved in 00 matches}}\\
\text{Cosine}(\boldsymbol{x},\boldsymbol{y}) &= \frac{\boldsymbol{x}^\top\boldsymbol{y}}{\|\boldsymbol{x}\|\|\boldsymbol{y}\|}\\
\text{ExtendedJaccard}(\boldsymbol{x},\boldsymbol{y}) &= \frac{\boldsymbol{x}^\top\boldsymbol{y}}{\|\boldsymbol{x}\|^2+\|\boldsymbol{y}\|^2-\boldsymbol{x}^\top\boldsymbol{y}}\\
\text{Correlation}(\boldsymbol{x},\boldsymbol{y}) &= \frac{\text{cov}(\boldsymbol{x},\boldsymbol{y})}{\text{std}(\boldsymbol{x})\text{std}(\boldsymbol{y})}
\end{aligned}$$

where $\text{cov}(\boldsymbol{x},\boldsymbol{y})$ denotes the covariance between $\boldsymbol{x}$ and $\boldsymbol{y}$ and $\text{std}(\boldsymbol{x})$ denotes the standard deviation of $\boldsymbol{x}$.

Notice that the SMC and Jaccard similarity measures only are defined for binary data, i.e., data that takes values of $\{0,1\}$. As the data we analyze is non-binary, we will transform the data to be binary when calculating these two measures of similarity by setting

$$x_i = \begin{cases} 0 & \text{if } x_i < \text{median}(\boldsymbol{x}) \\ 1 & \text{otherwise.} \end{cases}$$

3.2.1 Inspect and run the script `ex3_2_1.m`. The script loads the CBCL face database, computes the similarity between a selected query image and all others, and display the query image, the 5 most similar images, and the 5 least similar images. The value of the used similarity measure is shown below each image. Try changing the query image and the similarity measure and see what happens.

3.2.2 We will investigate how scaling and translation impact the following three similarity measures: Cosine, ExtendedJaccard, and Correlation. Let α and β be two constants. Which of the following statements are correct?

$$\begin{aligned}
\text{Cosine}(\boldsymbol{x}, \boldsymbol{y}) &= \text{Cosine}(\alpha\boldsymbol{x}, \boldsymbol{y}) \\
\text{ExtendedJaccard}(\boldsymbol{x}, \boldsymbol{y}) &= \text{ExtendedJaccard}(\alpha\boldsymbol{x}, \boldsymbol{y}) \\
\text{Correlation}(\boldsymbol{x}, \boldsymbol{y}) &= \text{Correlation}(\alpha\boldsymbol{x}, \boldsymbol{y}) \\
\text{Cosine}(\boldsymbol{x}, \boldsymbol{y}) &= \text{Cosine}(\beta + \boldsymbol{x}, \boldsymbol{y}) \\
\text{ExtendedJaccard}(\boldsymbol{x}, \boldsymbol{y}) &= \text{ExtendedJaccard}(\beta + \boldsymbol{x}, \boldsymbol{y}) \\
\text{Correlation}(\boldsymbol{x}, \boldsymbol{y}) &= \text{Correlation}(\beta + \boldsymbol{x}, \boldsymbol{y})
\end{aligned}$$

Hints:

- *Inspect and run the script* `ex3_2_2.m` *to check your answers.*
- *Type* `help similarity` *to learn about the Matlab function that is used to compute the similarity measures.*
- *Even though a similarity measure is theoretically invariant e.g. to scaling, it might not be exactly invariant numerically.*

3.3 The Univariate and Multivariate Normal Distribution

3.3.1 Generate 200 samples from a Normal distribution with mean $\mu = 17$ and standard deviation $\sigma = 2$. Plot the samples as well as a histogram of the samples.

Hints:

- *Type* `help normrnd` *to learn how to generate Normal distributed random numbers in Matlab.*
- *The function* `plot` *can be used to plot the samples.*
- *The function* `hist` *can be used to plot a histogram.*
- *Type* `help hist` *to see how to change the number of bins.*
- *If you are stuck, take a look at the solution in* `ex3_3_1.m`*.*

Try running the code a few times and see how the data and histogram changes when new random numbers are generated from the same distribution.

The histogram is generated by counting how many of the samples fall within the range covered by each bin of the histogram. Try changing the number of bins in the histogram.

3.3.2 Compute the empirical mean and standard deviation of the generated samples. Show, that they are close but not equal to the theoretical values used to generate the random samples.

Hints:

- *Take a look at the functions* `mean` *and* `std`*.*
- *If you are stuck, take a look at the solution in* `ex3_3_2.m`*.*

Try running the code a few times and see how the empirical mean and standard deviation changes when new random numbers are generated from the same distribution.

In the next part we will see how the number of samples influence the resulting histogram.

3.3.3 Inspect and run the script `ex3_3_3.m` which generates random samples from the Normal distribution and plots the histogram as before. Also, the function plots the true probability density function of the Normal distribution.

Show that when the number of samples `N` is increased, the histogram approximates the pdf better and the empirical estimates of the mean and standard deviation improve.

So far we have been considered a 1-dimensional Normal distributed random variable. In the next step we will consider the multivariate Normal distribution in two dimensions.

The covariance matrix for a 2D Gaussian is described by

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1^2 & \text{cov}\left(x_1, x_2\right) \\ \text{cov}\left(x_2, x_1\right) & \sigma_2^2 \end{bmatrix},$$

where $\text{cov}(\cdot, \cdot)$ is covariance between two random variables. Note that $\text{cov}\left(x_1, x_2\right) = \text{cov}\left(x_2, x_1\right)$, i.e., the covariance matrix is symmetric, and $\sigma_n^2 = \text{cov}\left(x_n, x_n\right)$. We can write the covariance matrix in terms of Correlation, i.e.

$$\begin{aligned} \text{Correlation}\left(x_1, x_2\right) &= \frac{\text{cov}\left(x_1, x_2\right)}{\sqrt{\text{cov}\left(x_1, x_1\right)}\sqrt{\text{cov}\left(x_2, x_2\right)}} \Rightarrow \\ \text{cov}\left(x_1, x_2\right) &= \text{Correlation}\left(x_1, x_2\right)\sqrt{\text{cov}\left(x_1, x_1\right)}\sqrt{\text{cov}\left(x_2, x_2\right)}. \end{aligned}$$

3.3.4 Generate 1000 samples from a 2-dimensional Normal distribution with mean

$$\boldsymbol{\mu} = \begin{bmatrix} 13 & 17 \end{bmatrix}$$

and covariance matrix

$$\boldsymbol{\Sigma} = \begin{bmatrix} 4 & 3 \\ 3 & 9 \end{bmatrix}.$$

Hints:

- *Type* `doc mvnrnd` *to learn how you can generate multivariate Normal distributed random numbers in Python.*

· *If you are stuck, take a look at the solution in* `ex3_3_4.m`.

3.3.5 Inspect and run the script `ex3_3_5.m` which generates 2-dimensional Normal distributed random samples and plots a scatter plot as well as a 2-dimensional histogram. In the script, the covariance matrix is constructed from the standard deviations and correlation as described above.

Show that when the correlation between x_1 and x_2 is zero, the scatter plot and 2-d histogram have the shape of an axis-aligned ellipse. Can you explain why?

Show that when the correlation between x_1 and x_2 is one, the values of x_1 and x_2 fall on a straight line. Can you explain why?

Try varying the number of samples, the mean, the standard deviations, the correlation and the number of histogram bins and see what happens.

3.4 Statistics on Digits

In this part we will use the concepts and commands from the previous section in order to make some very simple statistical models of the digits data set (16×16 pixel images of hand written digits) we considered in last weeks exercise.

3.4.1 Inspect and run the script `ex3_4_1.m`. The script loads the digits data set, and computes the mean of each pixel, the standard deviation of each pixel, and the covariance matrix between the pixels. By default, only images of "ones" are included in the analysis.

Does the mean look like you expect? Can you explain why the standard deviation is highest along the edges of the digit one? Try to change the digit you analyze and get a felling of how different the individual digits are.

For each pixel we now have a mean and a standard deviation, i.e., 256 means and 256 corresponding standard deviations. So in essence we can make a a simple model of the digits with a Normal distributions for each individual pixel.

Since we know how to draw a new sample from a Normal distribution we can draw a sample for each individual pixel based on their respective 1D normal distribution (i.e. draw a total of 256 values). Combining these samples we end up with a new digit. The question is now how natural our newly generated/artificial samples are, and if they at all are possible to recognize as digits.

With our simple model above we argued that we had 256 different 1D Normal distributions; however, we could also look at the problems in terms of the multivariate Normal. Instead of assuming 256 independent 1D Gaussians we could formulate our model for digits as a 256-dimensional multivariate Normal, which allows each pixel to depend on the other pixels. This dependency is described through the covariance matrix.

3.4.2 Inspect and run the script `ex3_4_2.m`. The script generates 10 new images with the same mean and standard deviation as the data using a 1-dimensional Normal distribution for each pixel. Next, the script generates 10 new images with the same mean and covariance as the data using a $16 \cdot 16 = 256$-dimensional multivariate Normal distribution.

Which model is best? Try changing the analyzed digits and see what happens.

3.5 Extra challenge

3.5.1 Try to vary the number of observations of a given digit you use to estimate the mean and (co)variance. Does the number of observations used make a difference on the quality of the generated digits?

3.6 Tasks for the report

After today's exercise you should be able to calculate the relevant summary statistics for the attributes of your data and evaluate the extent to which attributes are correlated with each other, see also the function `corrcoef.m`.

References

[1] Tamara L Berg, Alexander C Berg, Jaety Edwards, and DA Forsyth. Who's in the picture. *Advances in neural information processing systems*, 17:137–144, 2005.